INAUGURAL - DISSERTATION

zur

Erlangung der Doktorwürde

der

Naturwissenschaftlich-Mathematischen Gesamtfakultät

der

Ruprecht - Karls - Universität

Heidelberg

vorgelegt von

Diplom-Ingenieur (BA), M.Sc.	Frank Noé
aus:	Zweibrücken
Tag der mündlichen Prüfung:	27.01.2006

Thema

Transition Networks

Computational Methods for the Comprehensive Analysis of Complex Rearrangements in Proteins

---

Gutachter: Prof. Dr. Gerhard Reinelt
Prof. Dr. Jeremy C. Smith

Zusammenfassung

Strukturelle Umordnungen sind essenziell für die biologische Funktion von Proteinen. Bei solchen Umordnungen handelt es sich oft um komplexe Zustandsübergänge, die durch eine Vielzahl von Pfaden durch einen hochdimensionalen Konformationsraum charakterisiert sein können. Bisher sind keine Experimente verfügbar, die mögliche Mechanismen solcher Übergänge identifizieren können. Direkte Computersimulationen der Proteindynamik sind dazu ebenso ungeeignet, da die gegenwärtig erreichbare Simulationszeit mehrere Größenordnungen unter der typischen Zeitdauer komplexer Übergänge liegt. In dieser Arbeit wird ein Divide-and-Conquer Ansatz basierend auf Transition Networks (TN) vorgestellt. Ein TN ist ein gewichteter Graph, welcher die experimentell bestimmten Endzustände durch ein dichtes Netzwerk von Teilübergängen (der Kanten) über Zwischenzustände niedriger Energie (der Knoten) verbindet.

Es wird gezeigt, wie die Generierung und Analyse von TN, die bisher nur für kleine Polypeptide möglich war, für Proteine durchgeführt werden kann. Zur Erzeugung der TN Knoten wird eine effiziente hierarchische Methode entwickelt. Diese generiert eine gleichförmig verteilte Menge von Protein-Konformationen in einem für den Zustandsübergang relevanten konformationellen Unterraum. Die Bestimmung der TN Kantengewichte ist sehr berechnungsaufwändig. Hierzu wird ein graphentheoretischer Ansatz vorgestellt, der es ermöglicht, globale Netzwerkeigenschaften zu bestimmen, wobei lediglich die Werte einer kleinen Untermenge von Kantengewichten tatsächlich ermittelt werden müssen. Auf diesem Ansatz basierend werden Algorithmen angegeben, welche die besten Pfade des Übergangs sowie die Energiegrate zwischen den Endzuständen berechnen.

Die hier vorgeschlagene Vorgehensweise wird auf den konformationellen Schalter des Proteins Ras p21 angewandt. Die 32 besten Übergangspfade mit Ratenbestimmenden Energiebarrieren von bis zu 15 kcal/mol über dem besten Pfad werden ermittelt. Weiterhin werden die zwei wichtigsten Energiegrate zwischen den Endzuständen bestimmt. Diese sind jeweils mit der Umordung der Switch I und Switch II Bereiche im Protein assoziiert. Basierend auf den Ergebnissen werden drei konkurrierende Mechanismen für den Übergang von Switch I identifiziert. In all diesen Mechanismen bewegt sich die Seitenkette von Tyr32 unterhalb des

Proteinrückgrates, danach erfolgt der Raten-bestimmende Übergang von Switch II. Die Entfaltung der Switch II Helix folgt in allen möglichen Pfaden einem ähnlichen Muster und verläuft vom N-terminalen zum C-terminalen Ende hin. Trotz dieser Gemeinsamkeiten unterscheiden sich die zugänglichen Übergangpfade hinsichtlich der genauen Abfolge und der detaillierten Realisierung der konformationellen Ereignisse. Dies zeigt, dass komplexe Zustandsübergänge in Proteinen tatsächlich durch strukturell verschiedene Pfade realisiert werden können.

Wie die Anwendung auf Ras p21 demonstriert, können die hier vorgestellten Methoden dazu dienen, sehr komplexe Mechanismen in Proteinen, unabhängig von deren Zeitdauer, aufzuklären. Dies ist ein signifikanter methodischer Fortschritt im Bereich der molekularen Biophysik.

Abstract

Structural rearrangements in proteins are essential for biological function. Often, these are complex transitions, involving a multitude of pathways through a high-dimensional conformational space. As yet, no experiments are available to identify the possible mechanisms of these transitions. Direct computer simulations of protein dynamics can neither be used, as the simulation time presently accessible to them is several orders of magnitude below the timescale on which complex transitions occur. In the present work, a divide-and-conquer approach based on Transition Networks (TN) is proposed. TN are weighted graphs, which connect the experimentally determined end-state structures by a dense network of sub-transitions (the network edges) *via* low-energy intermediates (the network vertices).

It is shown here how the computation of TN, previously feasible only for small polypeptides, can be achieved for a protein. To generate the TN vertices, an efficient hierarchical procedure is developed which uniformly samples the conformational subspace relevant to the transition. As the determination of TN edge weights is computationally very expensive, a graph-theoretical approach is presented here which allows global network properties to be determined while only having to compute a small subset of edge weights. Following this approach, algorithms are presented to compute the best path connecting, and the energy ridge separating the transition end-states.

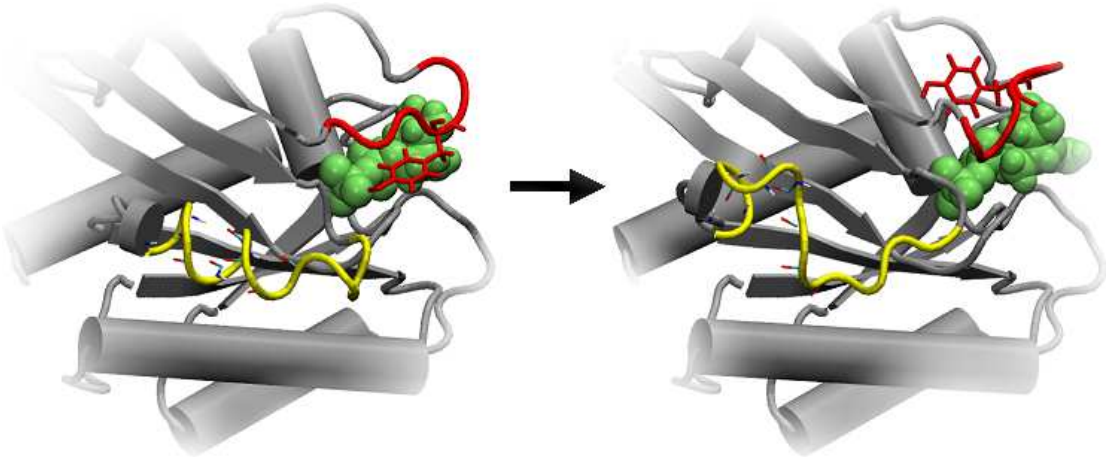
The approach is illustrated on the conformational switch of Ras p21. The 32 best transition pathways with rate-limiting barriers up to 15 kcal/mol above the globally-best pathway were determined, as well as the two main energy ridges, which involve rearrangements of the Switch I and Switch II loops, respectively. Based on these results, three competing pathways for the rearrangement of Switch I were identified, in all of which Tyr32 is threaded underneath the protein backbone. Subsequently, the rate-limiting unfolding of Switch II occurs, which follows a similar pattern among the best paths and progresses from the N-terminal to the C-terminal end. Despite these similarities, the precise order and the detailed realization of conformational events in Switch I and II varies, showing that complex conformational transitions in proteins may indeed occur *via* multiple pathways.

As the Ras p21 application demonstrates, the methodology developed here is useful to understand very complex mechanisms in proteins independent of their typical timescale. This represents a significant methodological progress in the field of molecular biophysics.

Contents

Chapter 1

Introduction



Life emerges from a complex network of interactions between agents on multiple levels of hierarchy: organisms, organs, cells, organelles and biomolecules. On the smallest scale, proteins and other biomolecules serve as nanomachines that are specialized to perform particular tasks involving communication, transport, storage, chemical modification or mechanical work. Proteins are complex, dynamical systems, consisting typically of several thousand atoms. Most proteins are able to 1) self-organize through the process of *protein folding* from the unfolded state in which they are manufactured into the *native state* in which they perform their task, and to 2) switch among a finite number of native sub-states. These *conformational changes* are ubiquitous processes and are critical for biological function. For example, the cooperative rearrangement of the Hemoglobin subunits between the oxygen-bound and -unbound states (Fig 1.1A) allows for an efficient uptake of oxygen in the lungs and its transport to the muscles [1, 2]. The contraction of the muscle is based on the relative sliding motion of filaments that consist of the proteins actin and myosin. This sliding motion is caused by a conformational change in the myosin proteins, called *power-stroke*, which rotates the myosin head relative to a lever arm [3, 4] (Fig 1.1B). As a third example, the growth and reproduction of cells is a complex process which involves the coordination of the cooperative work of several thousand different types of proteins. The Ras p21 protein carries a binary state of information that is communicated to other proteins involved in the process. The transition between the active and inactive forms of Ras, called *molecular switch* (Fig 1.1C), is an essential control for cell metabolism and is strongly related to the occurance human tumors [5, 6]. A detailed understanding of the mechanism of these transitions is interesting from the theoretical point of view and has a high potential impact on medical and biotechnological applications.

Like most conformational transitions, the above examples are *thermally activated*. This involves that the end-states of the transition (the *reactant* and *product* structures) are both metastable, *i.e.* if the protein is in one of these states, it remains there on a relatively long timescale (typically up to microseconds), until a sufficiently strong thermal activation carries it out of that state. This allows to design experiments using X-ray crystallography and nuclear magnetic resonance spectroscopy that provide atomic-detail structures of the end-states, and sometimes long-lived intermediates, of conformational transitions [7].

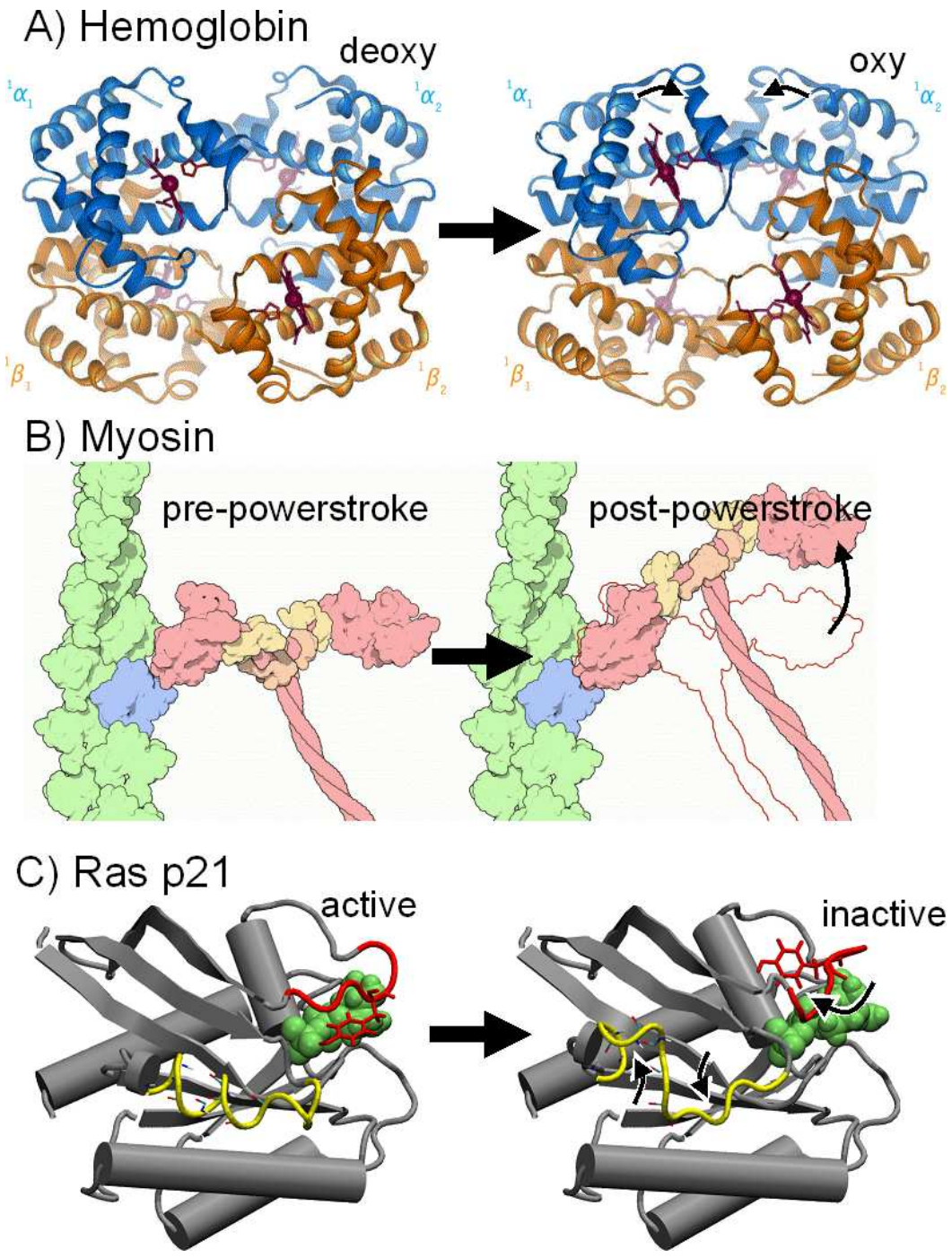


Figure 1.1: Well-known examples for complex conformational changes in proteins. A) The cooperative rearrangement of the subunits in the hemoglobin tetramer upon oxygen uptake. B) The power-stoke in myosin (red), when it is attached to actin (green). C) The conformational switch of Ras p21 from the active to the inactive form. Most changes occur in the Switch I (red) and Switch II (yellow) regions.

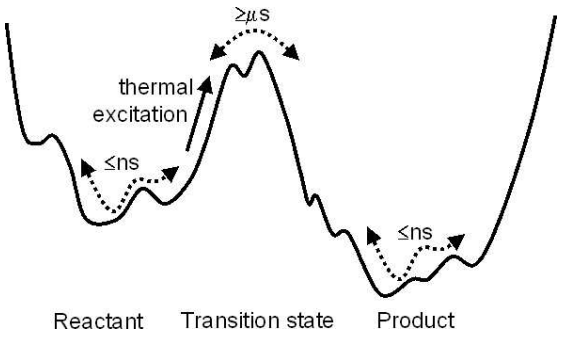


Figure 1.2: Relationship between the energy barrier of a transition and the time required for it. In the nanosecond time scale, biomolecules diffuse and vibrate in their stable end-states. Occasionally (microsecond time scale), sufficient thermal energy is accumulated to overcome the transition barrier.

The probability p to find the protein in a particular conformation is related to the energy E of that conformation *via* the Boltzmann distribution $p \propto \exp(-E/k_B T)$, where k_B is the Boltzmann constant and T is the temperature. This implies that stable states, such as the transition end-states are characterized by energy basins. A thermally activated process is the departure of the system out of such an energy basin by overcoming an energy barrier through random thermal excitation (see Fig. 1.2). Given constant temperature T, the probability of such an event per unit time decreases exponentially with increasing barrier height. Conformational changes in proteins typically involve barriers such that one transition event is expected to happen on a timescale of microseconds or longer. The activated transition states involved in such a transition, however, are very short lived as the system quickly relaxes towards lower-energy regions. Because of this transition-state instability, the structural mechanism of transitions can normally not be experimentally resolved. Complex or large-scale conformational transitions pose a particular challenge because their mechanism (*i.e.*, the order and nature of their sub-transitions) is difficult to predict and may, in principle, occur *via* various pathways. Computer simulation can help to gain insight into these mechanisms.

The state-of-the-art approach to model proteins is through atom-based models where the interatomic interactions are defined by an empirical potential energy function. The energy function is typically a sum of bonded and non-bonded interaction terms. Bonded interaction terms account for deformation of bond lengths (distances between covalently bonded atom pairs), valence angles (angles between covalently bonded atom triples) and dihedral angles (torsion angles between covalently bonded atom quadruples). Nonbonded interaction terms evaluate electrostatic interaction between charged atoms and van-der-Waals interactions (most

importantly steric clashed between nonbonded atom pairs below some distance). See Fig. 1.3A for an overview of the typical energy terms. Adding up these terms for all interacting atom-pairs in a given protein yields the empirical energy function. This energy function contains many parameters (*e.g.* standard bond lengths, bond stretching force constants, atomic charges), which have been determined by fitting simulation data obtained by using the energy function against data from experiments or quantum-mechanical *ab initio* calculations. Several molecular simulation packages, such as Charmm [8], Amber [9], or Gromos [10], include a definition of the energy functional terms and also deliver parameters for the atom types that typically occur in biomolecules. By specifying the topology of a protein (*i.e.* which atoms are contained in the protein and how are they bonded), the energy function for this particular protein is defined and can be used as a computational model.

The energy function assigns a value $E_{\mathrm{pot}}(\mathbf{x})$ to each molecular configuration $\mathbf{x}$, *i.e.* to each atom position vector. For a N-atomic system, a configuration is defined by $3N$ coordinates. Therefore $E_{\mathrm{pot}}(\mathbf{x})$ defines an energy surface on a $3N$-dimensional hyperspace. For proteins in the native state, this *potential energy surface* (PES) is known to be rough and to contain a vast number of minima and saddle points. Fig. 1.3B shows a scheme of such a potential energy surface.

A rigorous computer simulation method to explore the dynamics of the protein on the energy surface is *molecular dynamics*. Initial conditions are defined by setting the atomic coordinates according to one of the experimentally known transition end-states of the protein, say $\mathbf{x}_R$, and the velocities $\mathbf{v}$ according to a distribution that yields a desired overall temperature T. One then integrates a system of (possibly stochastic) differential equations (*e.g.* Newtonian dynamics or Langevin dynamics) and thereby follows the trajectory of the system through phase-space. The limitation of this approach is that the presently accessible simulation time is in the range of 10 to 100 nanoseconds, given the complexity of the calculations and the allowable length for an integration time step. Since complex conformational transitions typically occur on a timescale of microseconds or more (see Fig. 1.2), the simulation time is much too short to observe even a single transition event. This is an aspect of the well-known *sampling problem*.

Variations of molecular dynamics have been proposed to overcome this timescale

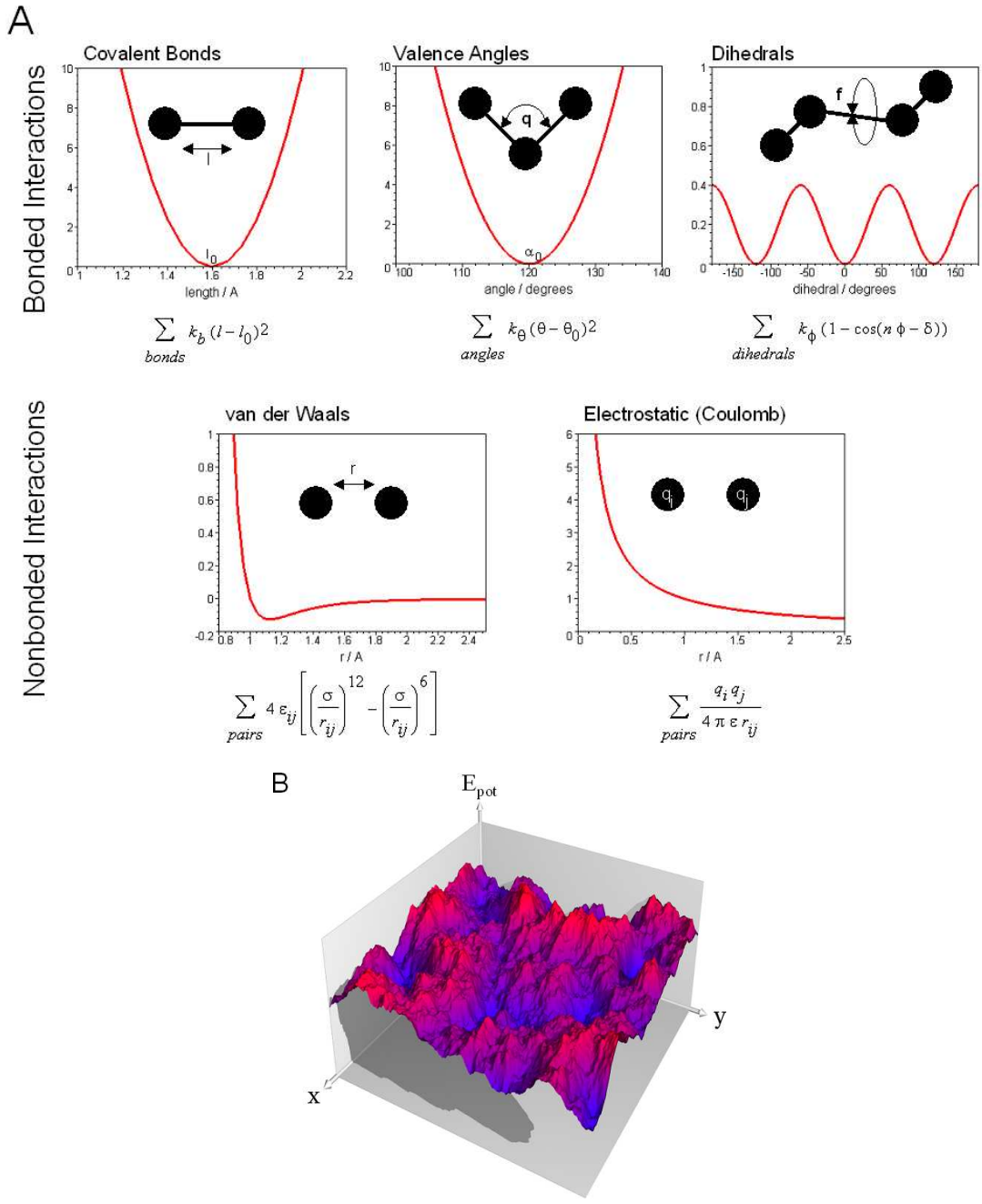


Figure 1.3: Empirical energy function and potential energy surface defined by it. A) Typical terms in the empirical energy function that is used to model biomolecules. Covalent bond lengths and valence angles are described as harmonic springs. The periodic torsional potentials are modeled by sine functions. Torsions around covalent single bonds involve low-energy barriers and yield very flexible degrees of freedom. Steric repulsion and dispersive attraction between non-bonded atoms is modeled with the van-der-Waals interaction. Atoms carrying partial charges may attract or repel each other, as described by a Coulomb term. B) The energy function terms add up to a function that maps a $3N$-dimensional coordinate vector to a real-valued potential energy. It therefore defines a potential energy surface (PES) on a $3N$-dimensional hyperspace that is very rough, containing a vast amount of minima and saddle-points.

problem. For example, multiple time-step methods [11] are quite successful in other multiple-timescale contexts but do not achieve sufficient speedup for the present purposes [12, 13]. Other methods bias the underlying energy potential [14] or reduce the dimensionality of the conformational space [15]. These methods face the difficulty that a good guess of the energy surface along the whole transition must in principle be known *a priori*, which is usually not possible for complex transitions in proteins. Steered and targeted molecular dynamics [16, 17] incorporate a constraint into the energy function that directs the system toward the desired product structure. While these methods are successful in cases where the transition follows a pathway that is compatible with these constraints [18], they lead to unnatural structures and unrealistic energy barriers in other cases [19]. A further variant is conformational flooding [20], which approximates the local shape of the underlying energy surface explored by a molecular dynamics trajectory by computing its main directions of motion and then escapes the local energy minimum by adding a corresponding multivariate Gaussian function to the energy function. Although this method allows the trajectory to overcome high energy barriers, it does not necessarily yield the desired transition.

Pathway methods are a different approach to simulating molecular transitions. Starting from an initial guess of the transition pathway, the latter is allowed to relax on the energy surface by constrained molecular dynamics [21] or by local minimization methods [22, 23, 24]. These methods have been applied successfully in cases where the transition does not involve too complex rearrangements of the protein, such that a number of reasonable initial guesses of the pathway can easily be formulated [25, 26, 27]. However, when the transition involves rearrangements of the protein fold, a guess for the initial path is more difficult to make. Moreover, such transitions can follow multiple pathways, as the energy landscape is likely to include broad energy ridges with many saddle-points of similar energies. Therefore, the determination of a single reaction pathway (even if it is the lowest-energy one) is not sufficient to fully describe the transition [19].

To represent multiple pathways, the *Transition Network* approach may be used, which is formulated in the present work. Transition Networks (TN) are a discrete and simplified representation of configurational space. Following a "divide and conquer"-strategy they encode the possible transition pathways in a network of sub-transitions. Each sub-transition occurs between two conformations that are

relatively close in conformational space. Each conformation in the network can be reached and left through at least one, but usually several, sub-transitions. Each sub-transition has an associated energy barrier that can be used to determine a rate constant or a mean passage time (*i.e.* "cost") for it.

The construction of Transition Networks is documented in a large number of studies which have addressed the analysis of energy surfaces by mapping their local minima and saddle points [28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49]. These stationary points can be generated by local optimization starting from conformational ensembles that are generated by high-temperature molecular dynamics [32, 36, 39, 42, 50], by a mode-following guided parallel search starting from a deep initial minimum [38, 45, 51], or by Discrete Path Sampling (DPS) [46, 48, 52]. The kinetics between groups of stationary points may be recovered using Master-Equation dynamics (MED) [31, 32, 35, 38, 39, 41, 42, 43, 45, 46, 48, 49], Kinetic Monte Carlo (KMC) [48], or, again, by Discrete Path Sampling (DPS) [46, 48, 52]. Typical applications of the above methodology are the rearrangement of atomic or molecular clusters [32, 33, 34], the rearrangement or folding of peptides [30, 31, 35, 36, 38, 39, 41, 43, 46, 48] and of model proteins [37, 45, 53].

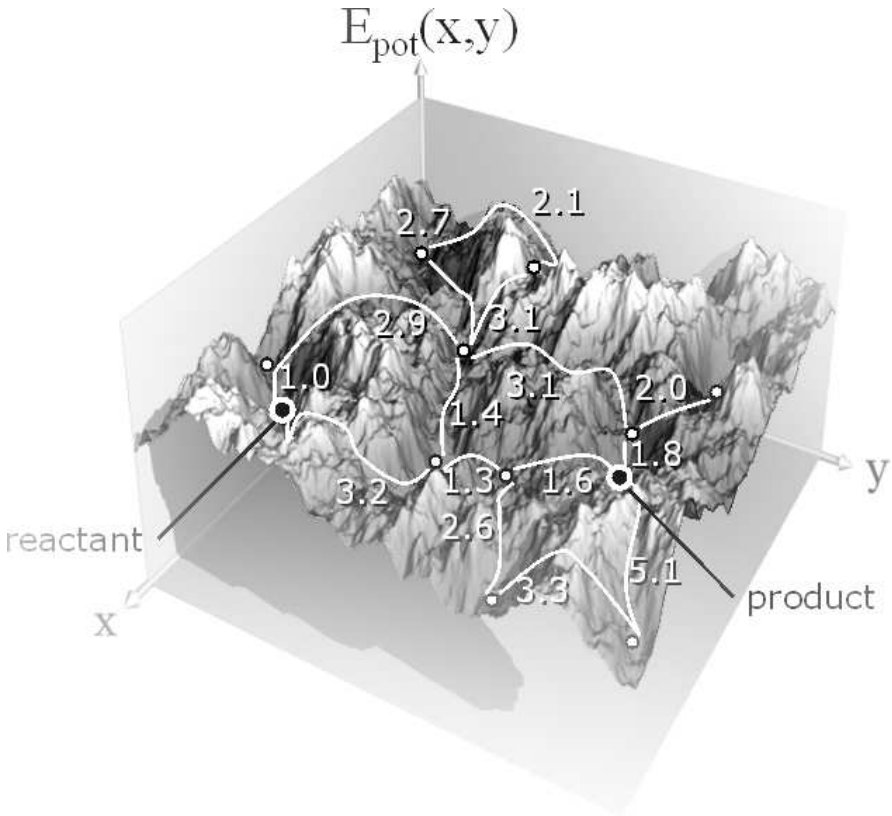
The applicability of these approaches to complex transitions between native conformations of a protein is limited by two main difficulties. The first involves the generation of the minima which serve as TN vertices: It is *a priori* unclear how a conformational ensemble can be generated that adequately covers the volume of conformational space that is relevant for the transition. In particular, the direct manipulation of the backbone torsion angles or high-temperature dynamics are likely to disrupt the native structure, while search-based procedures may get lost in the huge number of possibly distant low-energy minima. Discrete Path Sampling is likely to be successful in identifying a connected channel between the end-states, but it is unclear how it can identify a collection of considerably different channels. The second problem involves the computation of energy barriers. The determination of global properties of the network, such as the kinetics or the optimal path between two end-states [54], requires the barriers of the sub-transitions in the network to be known. Dense Transition Networks for complex macromolecular transitions typically have so many edges and the computation of each sub-transition barrier is so CPU-demanding, that the computation of all

sub-transition barriers cannot be afforded. Both problems are addressed in the present work.

The thesis is organized as follows: Chapter 2 provides a theoretical framework of Transition Networks. It relates the theory of Transition Networks to other molecular simulation approaches and lays down a computational formalization for the biophysical questions relevant to this work. Chapter 3 presents a graph-theoretical approach that allows to determine global network properties (such as the best transition pathway or the dividing energy ridge) based on the computation of only a limited subset of all sub-transition barriers. This allows the Transition Network approach to be applied to very complex transitions, involving networks with $O(10^4)$ to $O(10^6)$ edges. While the methodology developed to that point can be used to generate and analyze Transition Networks of arbitrary molecular processes, the subsequent chapters are dedicated to the specific case of proteins. Chapter 4 presents a procedure for efficiently sampling the relevant degrees of freedom of a complex transition in a protein, such as to determine the Transition Network vertices in these cases. In Chapter 5, the methods introduced here are applied to identify likely pathways and the order of events in an example system, the molecular switch of Ras p21. This chapter reports the first successful attempt to generate and analyze a comprehensive set of multiple pathways involved in a complex conformational transition of a protein. Chapter 6 concludes the work and proposes a number of promising follow-up studies.

Chapter 2

Theoretical Framework of Transition Networks



Transition Networks (TN) can in principle be used to model the kinetic behavior of any dynamical system that can be appropriately described by a (possibly large) number of states and interstate transition rules. This chapter provides a theoretical framework which establishes a connection between the potential energy surface of the system, the kinetics emerging from the dynamics on this surface, and the modeling steps required to formulate a network description of kinetic processes in the system. Methods are introduced which allow to analyze the Transition Network for global properties which predict aspects of the large-scale behavior of the system.

Assume that we are given a model for the system which maps a system configuration, or state vector $\mathbf{x}$ (here: the atomic coordinates) to a real-valued state property $E_{\mathrm{pot}}(\mathbf{x}) : \mathcal{R}^{\dim(\mathbf{x})} \rightarrow \mathcal{R}$ (here: the potential energy; $\dim(\mathbf{x}) = 3N$, where N is the number of atoms). When the system undergoes dynamical motion under specified conditions (*e.g.* Newtonian dynamics at some defined temperature), it samples configurations from a configurational state density $p(\mathbf{x})$. A Transition Network is a discrete representation of states and state-changes in the system, which abstracts local dynamical behavior and captures the system's relevant kinetic behavior. Formally, a Transition Network is equivalent to a weighted graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E}, ...)$ whose vertices, $\mathcal{V}$, represent states that are metastable under dynamical motion (*e.g.* energy minima or probability maxima) and whose edges, $\mathcal{E}$ represent sub-transitions between the states (*e.g.* saddle points or probability bottlenecks) of the system. Each vertex, $u \in \mathcal{V}$, corresponds to a group of system micro-states (here: a group of geometrically similar molecular configurations). It has a weight associated with it which quantifies that vertex' energy or probability. Each edge, $e = (u \in \mathcal{V}, v \in \mathcal{V}) \in \mathcal{E}$ represents the transition through a boundary surface separating two neighboring vertices. The weight associated with the edge quantifies its transition energy, rate, or mean passage time.

We distinguish between *static* and *dynamic* Transition Networks. Static TN describe features of the potential energy surface $E_{\mathrm{pot}}(\mathbf{x})$ and their network weights correspond to energies. Sec. 2.1 describes how static TN energies can be obtained. Dynamic TN incorporate thermodynamic and kinetic information, their network weights correspond to residence probabilities, transition rates or mean passage times. How to obtain these properties is described in Sec. 2.2. The remaining sections in this chapter concentrate on how global network properties

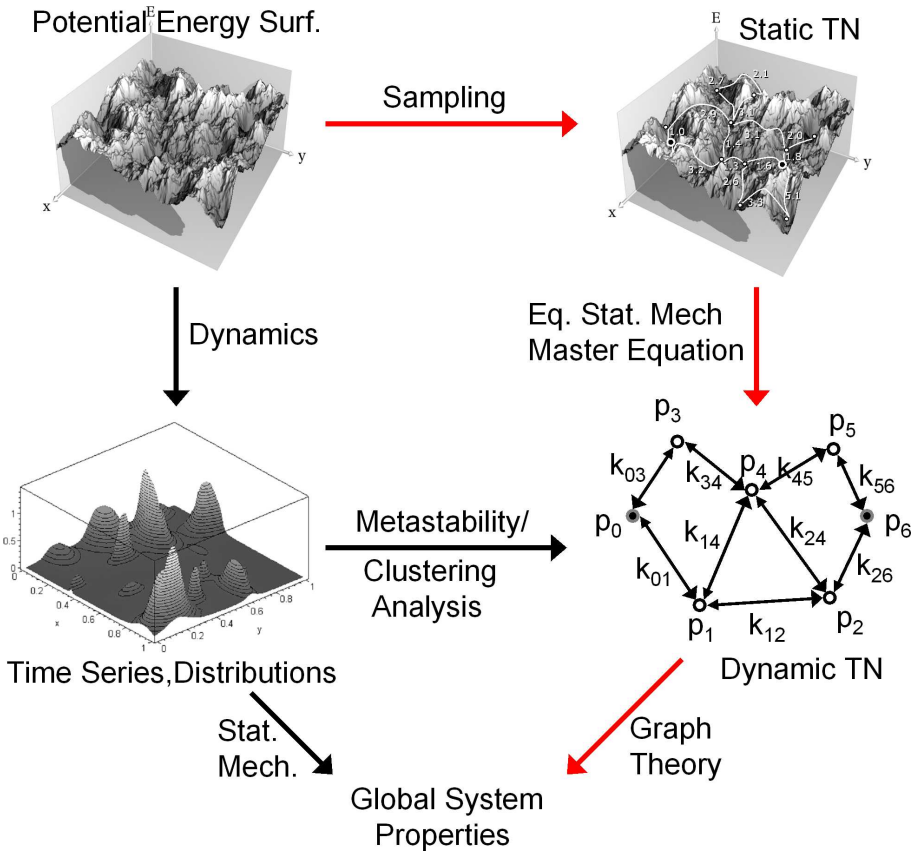


Figure 2.1: Illustration of the relationship of the Transition Network approach developed and followed in this work (red arrows) to other molecular simulation approaches (black arrows).

can be derived from either static or dynamic TN: best transition pathways (Sec. 2.3), energy ridges and rate-limiting transition surfaces (Sec. 2.4). As the theory which is appropriate to describe the relationship between static and dynamic TN depends on the class of the physical system that is described, the following sections concentrate on TN for molecular systems.

The general goal of molecular simulation is to compute some system properties which are, in general, global (*i.e.* they arise from the collective interplay of the microscopic interaction rules), such as the most dominant pathway for a transition between two defined system states or the mean time required for this transition. Fig. 2.1 illustrates how the present Transition Network approach is related to other approaches of molecular simulation. In the state-of-the-art procedure, the

system dynamics (*e.g.* classical) is computed based on the potential energy surface, giving rise to a time series (a trajectory through configurational space) and distributions (*e.g.* a configurational state density). Statistical mechanics is used to calculate the desired global system properties [55]. In the approach proposed here, one samples the potential energy surface and represents its features in a static TN. From this, a dynamic TN is generated using either equilibrium statistical mechanics (equilibrium case) or a master-equation approach (non-equilibrium case). Using graph theory paired with statistical mechanics allows to derive global system properties from the dynamic TN. Recent studies that are complementary to ours allow to derive dynamic TN from time series [56].

2.1 Static Transition Network Energies

In static TN, a vertex represents a region R of the configurational space, corresponding to an *attraction basin*. Given a procedure, minimize($\mathbf{x}$), which maps a state $\mathbf{x}$ by direct minimization (*e.g.* steepest descent) to a local minimum $\mathbf{x}_{\min}$, an attraction basin is defined as the union of configurations that converge to the same minimum [35]:

$$R(\mathbf{x}_{\min}) := \{\mathbf{x} | \text{minimize}(\mathbf{x}) = \mathbf{x}_{\min}\}.$$

Any given vertex v has associated with it the configuration of the corresponding minimum $\mathbf{x}_v$ and its energy E_v. The TN edges represent sub-transitions through the boundary regions separating adjacent vertices. They are therefore defined between pairs of neighboring vertices. Each edge $e = (u, v)$ is also associated with an edge energy E_{uv}. Fig. 2.2 shows a schematic representation of a static TN (vertex energies are not shown).

Ideally, E_v would correspond to the relative free energy of region R, $\Delta G_v = G_v - G_0$, where G_0 is some arbitrary reference energy. The edge energy E_{uv} should likewise correspond to the relative free energy of the transition state $\Delta G_{uv} = G^{\ddagger}_{uv} - G_0$. According to the first law of thermodynamics, free energy differences can be expressed as:

$$\Delta G = \Delta E_{\text{pot}} + \Delta E_{\text{kin}} + \Delta(pV) - T\Delta S, \tag{2.1}$$

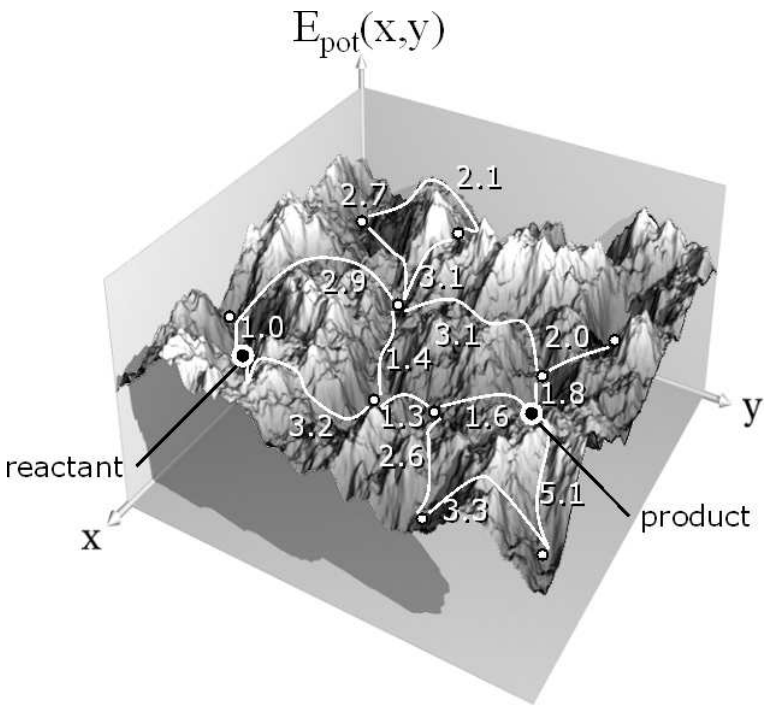


Figure 2.2: Static Transition Network on a schematic two-dimensional energy surface. The network vertices (white bullets) correspond to low-energy intermediates between the reactant and product end-states of the transition (black bullets). The network edges (white lines) correspond to sub-transitions between the vertices and are associated with the rate-limiting barrier energies along the sub-transitions (white numbers).

where E_{pot} is the potential energy, E_{kin} is the kinetic energy, p is the pressure, V is the volume, T is the temperature and S is the entropy. In liquid and solid systems and at low pressure, the pressure-volume product is nearly constant ($\Delta(pV) \approx 0$) [57]. Also, if the temperature and the number of particles are constant, $\Delta E_{\mathrm{kin}} = 0$, on average. Thus:

$$\Delta G \approx \Delta E_{\mathrm{pot}} - T\Delta S. \tag{2.2}$$

Accurate free energies are required for the static TN to derive reliable dynamic TN weights from these. Given current methodological and computational shortcomings, however, the calculation of reliable free energies is difficult, often even impossible. We therefore need to consider several levels of accuracy:

1. Constant-entropy approximation

   It is assumed that the entropic changes are negligible ($\Delta S \approx 0$) and the free energy difference is therefore approximated by the potential energy difference: $\Delta G \approx \Delta E_{\text{pot}}$.

2. Harmonic approximation

   The system is assumed to reside in the vicinity of a harmonic expansion around the minima and the paths of minimum energy. Free energy differences are given by an harmonic approximation:
   $\Delta G \approx \Delta E_{\text{pot}} - T\Delta S_{\text{harm}}$.

3. Free energy differences ΔG are computed.

The vertex and edge energies can be shifted by subtracting an arbitrary constant value E_0 without affecting the results. To avoid numerical problems when using exponentials of E_{uv}, it is desirable to keep E_{uv} small. Here this is done by choosing E_0 as the minimal vertex energy in the network.

The following subsections describe how vertex and edge energies, using the above three levels of precision, can be obtained. They also discuss the cases in which the different levels of precision are meaningful to be applied and when it is computationally and methodologically possible.

2.1.1 Constant-Entropy Approximation

In the constant-entropy approximation, we assume that the regions of configurational space corresponding to the different vertices are of approximately similar size and shape. We furthermore assume the transitions between them to lead through narrow reaction channels such that the transition-pathways are well defined. Following these assumptions, energy differences are dominated by enthalpic contributions, while the entropic contributions are comparatively small. Here, we set $\Delta S = 0$ in Eq. 2.2 which gives $\Delta G \approx \Delta E_{\text{pot}}$. The approximation is useful for systems with few degrees of freedom, having well-defined structures separated by high energy barriers, and which are studied at low temperatures. Biomolecules in physiological conditions (aqueous solvent, $T > 300K$) do not satisfy these conditions. For the dynamics of biomolecules, the constant-entropy approximation is

severe. When computing the weights for dynamic TN, errors on the energies are exponentially magnified. Therefore, this approximation disables any quantitative accuracy of the thermodynamics and kinetics and one has to confine oneself to qualitative conclusions based on correlations (see Sec. 2.2). The benefit of this approximation, is that the theory is very mature and it is always feasible to obtain potential energy differences ΔE_{pot} even for very large and complex systems.

Determination of the vertex energies E_u is trivial as it simply requires a local optimization of $E_{\mathrm{pot}}(\mathbf{x}_u)$ starting from some initial point $\mathbf{x}_{u,0}$ (The selection of these initial points is *not* trivial - this issue is addressed in chapter 4). The edge energy E_{uv} is given the value of the rate-limiting saddle point of the reaction channel connecting u and v. For this, we define a pathway of "least effort", *i.e.* one that can be accessed with a minimum amount of energy. Such a *Minimum Energy Path* (MEP) is a continuous path $\mathbf{z}(\lambda)$ connecting $\mathbf{x}_u$ and $\mathbf{x}_v$ ($\mathbf{z}(0) = \mathbf{x}_u$, $\mathbf{z}(1) = \mathbf{x}_v$, $\lambda \in [0,1]$), satisfying following criteria:

1. $\nabla E_{\mathrm{pot}}(\mathbf{z}(\lambda))|_{\perp} = 0 \; \forall \; \lambda \in [0,1]$, *i.e.* the gradient orthogonal to the path tangent is zero everywhere along the path.

2. $\mathbf{H}(\mathbf{z}(\lambda))|_{\perp}$ is positive definite $\forall \; \lambda \in [0,1]$, *i.e.* the Hessian matrix at each path-point, formulated in the subspace orthogonal to the path tangent, has only positive eigenvalues. Therefore, all path-points have a minimum of potential energy in all directions except the path tangent.

All local energy maxima along the MEP are first-order saddle points on $E_{\mathrm{pot}}(\mathbf{x})$ [58]. The highest-energy saddle point gives the transition state structure and the edge energy E_{uv}.

MEP are computed here with the Conjugate Peak Refinement (CPR) method [24]. CPR is an iterative method which is given as an initial pathway a set of points $P = [\mathbf{x}_u, \; \ldots, \mathbf{x}_v]$, which are *e.g.* generated by an interpolation between $\mathbf{x}_u$ and $\mathbf{x}_v$. In each iteration, CPR adds points to, removes points from or refines points in P, according to a heuristic set of rules, such as to refine the initial path to an MEP. The basic idea is to identify the points with maximum energy along the path (the "peaks") and to move these points closer to the MEP by a controlled conjugate gradient minimization in the complementary subspace (see Fig. 2.3). In

contrast to other MEP methods, such as Self-Penalty Walk [22], Nudged Elastic Band (NEB) [23] or the String Method [59], CPR automatically finds all saddle points along the path to a desired accuracy. The algorithm does not evaluate second derivatives, but uses only the energy (which must be continuous) and its gradient.

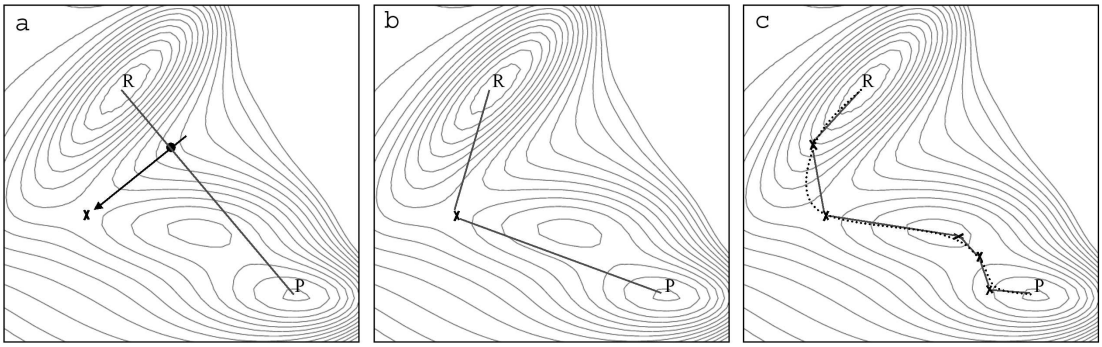


Figure 2.3: Illustration of the Conjugate Peak Refinement method. (a) Starting from an initial guess of the path (here: a linear interpolation between reactant (R) and product (P) states), the point of highest energy is found by maximization along the path (•). This point is moved closer to the MEP by a series of successive line-minimizations along directions conjugate to the path direction at (•). On the displayed 2D surface, this amounts to only one line-minimization (shown as →). (b) The optimized point (x) is inserted into the path. (c) This process of maximization/minimization is repeated until all local energy maxima along the path are identified as first-order saddle points. The path thus obtained (R-x-x-x-P) is a good approximation to the MEP (· · · · ·).

2.1.2 Harmonic Approximation

In the special case of the energy minima being deep and the reaction channel being a narrow pipe, the dynamical trajectories stay close to the minima and MEP. We therefore approximate the energy surface by a quadratic expansion around the minima and rate-limiting saddle points, assuming that the anharmonic portions of the energy surface (if any) are not sampled. This assumption is reasonable for local rearrangements in many solid state systems such as crystals, for most systems in the gas phase and for temperatures below the dynamical transition temperature the occurs in the range of 180 to 220 K [60]. Proteins in aqueous solvent at physiological temperature behave highly anharmonic, such that estimating their entropy based on a harmonic approximation cannot be expected

to be a considerable improvement over the constant-entropy approximation (Sec. 2.1.1) [61].

The application of the harmonic approximation is straightforward. Given a stationary point (minimum or saddle point), the vibrational frequencies can be obtained as the positive eigenvalues of a normal-mode analysis [62]. As the main contribution to entropy is given by the lowest-frequency motions, a full diagonalization of the Hessian matrix is often not necessary, so that a harmonic analysis can be conducted for very large systems [63].

Given the vibrational frequencies, or eigenvalues of the mass-weighted Hessian, ν_i, corresponding to the eigenvector representing the vibrational motion i, the vibrational entropy can be computed as[1]:

$$S_{\text{harm}} \approx k_B \ln \prod_{i=1}^{D} \frac{k_B T}{h\nu_i}, \tag{2.3}$$

where k_B and h are the Boltzmann and Planck constants, respectively, and T is the temperature. "Soft" potentials with low frequency motions allow to explore a wider range of configurations and therefore have a larger entropy than "stiff" potentials with high frequency motions[2]. The vertex energies E_u, in the harmonic approximation are given by substituting S_{harm} into Eq. (2.2):

$$E_u = E_{\text{pot}}(\mathbf{x}_u) - k_B T \ln \prod_{i=1}^{D} \frac{k_B T}{h\nu_{u,i}}.$$

To obtain the edge energy we compute the harmonic expansion at the rate-limiting saddle point of the MEP (see Sec. 2.1.1), where only the $D-1$ positive eigenvalues are considered. We obtain:

$$E_{uv} = E_{\text{pot}}(\mathbf{x}^{\ddagger}_{ub}) - k_B T \ln \prod_{i=2}^{D} \frac{k_B T}{h\nu^{\ddagger}_{uv,i}}.$$

[1]The precise formula is $S_{\text{harm}} = k_B \ln \prod_{i=1}^{D} z_i$, where z_i is the partition function of mode i: $z_i = [1 - \exp(-h\nu_i/k_B T)]$ [57]. For low frequencies ν_i, which contribute most to the vibrational entropy, we can use the approximation $\exp(x) \approx 1 + x$ [64], which leads to $z_i = k_B T/h\nu_i$.

[2]To be precise, the "stiffness" of the potential determines the frequency only if a mass-weighted potential is used, because of $\nu = \sqrt{k/m}$, where k is the force constant (second derivative of the unweighted potential along the vibrational mode) and m is the reduced mass of the mode.

2.1.3 Free Energies

Both approximations given above are generally not quantitatively valid for biomolecules in aqueous solution at physiological temperature. The dynamics of these systems often involves considerable changes in entropy and is not restricted to the harmonic regime near the energy minima. A rigorous treatment requires that vertex energy E_u is determined as free energy of the vertex region R_u and edge energy E_{uv} as the free energy of the boundary region between R_u and R_v, R_{uv}.

Free energy calculation methods, such as free energy perturbation or thermodynamic integration [55], attempt to compute free energy differences $\Delta G_{uv} = G_v - G_u$ between two thermodynamic states of the system by slowly changing one state into the other. Free energy barriers of transitions $\Delta G^{\ddagger}_{uv} = G^{\ddagger}_{uv} - G_u$ can be obtained with the umbrella sampling method [65, 66]. As any vertex u can serve as a reference point with $G_u = 0$, the static TN energies can be determined from free energy differences.

Two practical problems exist with this approach. Firstly, it must be assured that free energy calculations are confined to the regions R_u, R_v and to the reaction channel between them. To our best knowledge there is presently no free energy calculation method which would allow to use constraints of such a general type. Furthermore, free energy calculations on large systems such as proteins typically face the problem that convergence of the entropic contribution to free energy is very difficult to achieve. As a consequence, the present application of TN (Chapter 5) uses constant-entropy approximation (Sec. 2.1.1).

2.2 Dynamic Transition Network Weights

Dynamic TN describe the thermodynamics and the kinetics of the system. Vertex weights p_u correspond to the probability of finding the system in the state u, while edge weights K_{uv}, c_{uv} correspond to the rate (average number of transitions in unit time) and to the "cost" or effort of transition $u \to v$, respectively.

In the equilibrium, or stationary, case, the weights are by definition symmetric: $K_{uv} = K_{vu}$. In this case, which will be assumed throughout this work, the dynamic

TN weights can be directly obtained from the static TN weights using equilibrium statistical mechanics. The current section covers the computation of p_u, K_{uv} and c_{uv} for this case.

For a network which has not yet reached equilibrium, it is $K_{uv} \neq K_{vu}$. Networks with vertices that act as sources or sinks never reach equilibrium. In the general non-equilibrium case the dynamic TN weights are time-dependent: $p_u(t)$, $K_{uv}(t)$ and $\tau_{uv}(t)$ and global network properties which depend on these weights also change with time. The instantaneous network weights of non-equilibrium dynamic TN can be followed over time using a master-equation approach [35]. Because of the non-symmetric weights, time-dependent dynamic TN must be modeled with directed graphs, which can distinguish between edges (u, v) and (v, u).

As illustrated in Fig. 2.1, the structure and weights of dynamic TN are not necessarily based on static TN. In an approach that is similar to the sampling of the potential energy surface, dynamic TN can also be derived directly from dynamic data, such as molecular dynamics times series. Such an approach is followed in [56], where a hidden Markov model is generated that switches between discrete system states. Such a model is, in its spirit, similar to a dynamic TN. The difficulty with this route is that one first requires a time series in which the distribution of the global system property of interest has converged, and this is often difficult to obtain for large systems.

2.2.1 Vertex Weights p_u

The vertex weights p_u correspond to the probability of finding the system in state u. In the equilibrium case, this probability equals the fraction of the partition function associated with u [57]:

$$p_u = \frac{Z_u}{Z} = \frac{\exp\left(-\frac{E_u}{k_B T}\right)}{\sum_{v \in \mathcal{V}} \exp\left(-\frac{E_v}{k_B T}\right)}. \tag{2.4}$$

2.2.2 Sub-Transition Rates K_{uv} obtained with Transition State Theory

In a TN, the rate K_{uv} of a transition $u \to v$ is given as:

$$K_{uv} = p_u k_{uv}, \tag{2.5}$$

where p_u is the population or probability at state u and k_{uv} is the rate constant that captures the kinetic properties of that transition, such as the height and the form of the energy barrier.

To determine the rate constant k_{uv}, we apply Transition State Theory (TST). In TST, the reaction channel is partitioned into a reactant (R) and product (P) region, which are separated by a (possibly nonlinear) $(D-1)$-dimensional dividing surface, where D are the number of degrees of freedom in the system. Here, R and P correspond to two configurational regions R_u, R_v associated with vertices u and v and the dividing surface corresponds to the boundary between R_u and R_v. The TST equations are a result of "counting" the number of trajectories per unit time which pass across the dividing surface in the R→P direction. It relies on the following two assumptions [61]:

1) Identification of a perfect dividing surface. "Perfect" means, that no trajectory crosses the dividing surface twice. This requirement becomes irrelevant when one introduces a transmission coefficient $\kappa \in [0, 1]$ that is multiplied with the rate and is defined in such a way that it corrects the rate to account for re-crossings. If the dividing surface is placed such that the probability density of finding the system in the surface is minimal (usually at the highest energy ridge of the reaction channel), the number of re-crossings is low and κ is close to 1.

2) The transitions within the R region are assumed to be much more likely than the transition R→P. In this case, for each reactive trajectory, the reactant configurations can be assumed to have equilibrated and the likelihood of the transition can be computed taking into account only equilibrium properties of the R region and the dividing surface. This requirement is fulfilled if the energy barrier coinciding with the dividing surface is considerably higher than any energy barriers internal to R and also considerably higher than the thermal energy $k_B T$.

If both conditions are satisfied, TST applies. There are two equivalent formulations of the general TST equation for the rate constant, depending if they are derived from a phenomenological (macroscopical) or the statistical mechanical (microscopical) approach. Here, we choose the phenomenological formulation:

$$k_{\mathrm{TST}} = \gamma \exp\left(-\frac{\Delta G}{k_B T}\right), \tag{2.6}$$

where $\Delta G = G^{\ddagger} - G_R$ is the free energy difference between the transition state and the reactant region and the unspecified pre-factor γ contains dynamical effects, coming *e.g.* from recrossings, friction or viscosity. Using the nomenclature of TN, we can write for the rate constant of a transition from vertex u to vertex v:

$$k_{uv} = \gamma \exp\left(-\frac{E_{uv} - E_u}{k_B T}\right). \tag{2.7}$$

For the equilibrium case, we can combine this equation with Eqs. (2.4) and (2.5) and obtain the equilibrium rate as:

$$K_{uv}^{\mathrm{eq}} = \frac{\gamma}{Z} \exp\left(\frac{-E_u}{k_B T}\right) \exp\left(-\frac{E_{uv} - E_u}{k_B T}\right) = \frac{\gamma}{Z} \exp\left(-\frac{E_{uv}}{k_B T}\right). \tag{2.8}$$

The quality of the rate K_{uv}^{eq} depends on which level of accuracy is used to compute the edge energy E_{uv} (Sec. 2.1). As already denoted, using the constant-entropy approximation (Sec. 2.1.1) for proteins is such a severe approximation that one gives up quantitative accuracy at the level of rates. One may, however, still give a qualitative dependency of the rate on the (potential) edge energy:

$$K_{uv}^{\mathrm{eq}} \propto \exp\left(-\frac{E_{\mathrm{pot},uv}}{k_B T}\right), \tag{2.9}$$

which only states that the magnitude of the rate is proportional to the inverse exponential of the potential edge energy E_{uv} while entropic contributions are included in some unknown proportionality constant. Assuming K is dominated by $E_{\mathrm{pot},uv}$, there is a strong correlation between the two which allows us to give an approximate ranking of rates as a qualitative measure:

$$E_{\mathrm{pot},12} > E_{\mathrm{pot},34} \Rightarrow k_{12} \lessapprox k_{34}. \tag{2.10}$$

2.2.3 Mean Passage Times and Edge Costs

We use graph-theoretical algorithms to compute best paths (Sec. 2.3) which require having edge costs which are additive. In contrast to energies or rates, the mean passage time τ is an additive quantity. For a given transition, the mean passage time is simply the inverse of the rate:

$$\tau = K^{-1}. \tag{2.11}$$

Using the rate law in the form of Eq. 2.8, we can write:

$$\tau_{uv} = \frac{Z}{\gamma}\exp\left(\frac{E_{uv}}{k_B T}\right). \tag{2.12}$$

We define the edge costs as the normalized τ_{uv} which are obtained by setting the constant Z/γ to unity. The edge costs c_{uv} are therefore equal to the inverse Boltzmann weight of the edge energies:

$$c_{uv} = \exp\left(\frac{E_{uv}}{k_B T}\right). \tag{2.13}$$

As for the rates above, if the constant-entropy approximation is used to compute the edge energies, the edge costs must be understood as to yield a qualitative ranking of costs according to the energy.

$$\Delta E_{\mathrm{pot},12} > \Delta E_{\mathrm{pot},34} \Rightarrow c_{12} \gtrapprox c_{34}. \tag{2.14}$$

2.3 Best Paths

For a path connecting vertices $v_1 = v_R$ and $v_m = v_P$ *via* a series of m vertices, $P = (v_1, v_2, ..., v_m)$, travelling over edges $((v_1, v_2), ..., (v_{m-1}, v_m))$, the best path is defined as that which minimizes the cumulative edge costs

$$C(P) = \sum_{k=1}^{m-1} c_{v_k v_{k+1}}. \tag{2.15}$$

This definition of a best path is similar to the previously proposed notion of the continuous pathway with "maximum flux" or "minimum resistance" [67, 68]. To determine the best path in practice, the edge energies are transformed into a cost-vector $\mathbf{c}$ using Eq. (2.13). $\mathbf{c}$ has size $|\mathcal{E}|$ and assigns a cost c_{uv} to each edge (u, v) in $\mathcal{E}$. Subsequently, the Dijkstra algorithm [54] is used to identify a best path between the two end-states through the weighted network defined by $(\mathcal{V}, \mathcal{E}, \mathbf{c})$. This path minimizes the path cost $C(P)$ given in Eq. 2.15.

Because of the exponential weighting of energies in C, the best path tends to be one that minimizes the highest barrier along the path, *i.e.* it optimizes the rate-limiting step. Fig. 2.4 illustrates the concept of a best path through an energy surface.

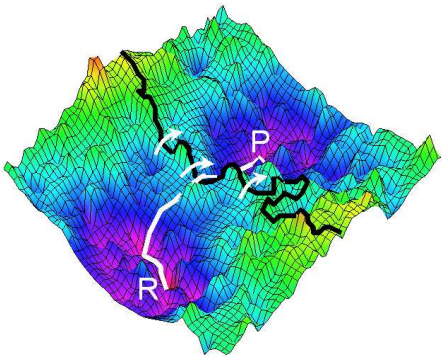


Figure 2.4: Illustration of a transition between two transition end-states (R and P) on a potential energy surface. The best path connecting R and P is shown as a white line. The energy ridge is shown as a black line. Alternative transition states crossing the energy ridge are shown as white arrows.

It is important to consider the limitations of the present definition of the best path. The edge costs, c_{uv}, used for this definition are based on the mean passage time, τ_{uv}, given in Eq. 2.12 which estimates the mean time that is expected between two passages over the barrier, based on the total number of passages per time unit. This quantity depends on the population on vertex u, p_u. The mean passage time is generally different from the average time that a single molecule needs to undergo the transition $u \to v$, as the latter does not depend on p_u. The best path, as it is defined here, measures the pathway of maximum traffic, but not the pathway for which a single molecule requires the least time. Fig. 2.5 shows an example where these two definitions of a best path differ. If comparison with experimental data is desired, an appropriate definition must be applied. In most cases, however, the precise definition of the best pathway is not critical, as for all reasonable definitions the highest energy barrier are the rate-limiting steps for the transition.

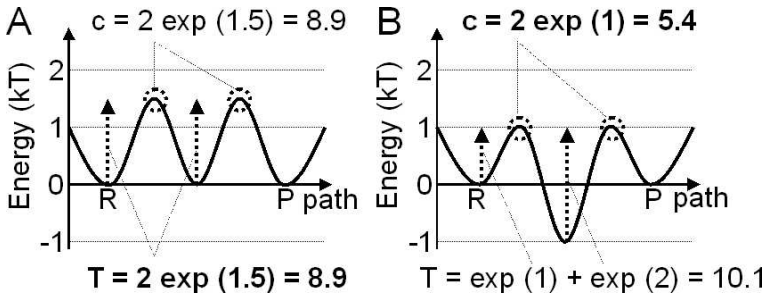


Figure 2.5: Illustration of the different meaning of the path cost C, which only takes into account the transition-state energies, and the normalized mean passage time τ_0 of a single molecule along the path, which is also affected by the depth of intermediates. According to the path cost C, the red path is preferred, while according to the mean passage time of the single molecule, the black path is preferred (bold).

A single best path dominates the transition only if the barriers of alternative pathways are considerably higher. However, the best path furnishes a preliminary understanding of the transition [19] or may be used as a guess for a reaction coordinate for free energy calculations [61]. A more extensive picture of the transition is given by computing multiple best paths (Sec. 2.3.3) or the energy ridge (Sec. 2.4).

2.3.1 Dijkstra Algorithm

The path connecting two network vertices v_R and v_P that minimizes Eq. (2.15) can be identified using the Dijkstra algorithm [54]. The algorithm actually identifies a whole tree of best paths from a given source vertex v_R to each other vertex v. If desired, the algorithm can be terminated as soon as the best path to the target vertex v_P is determined. This is not done here as the CPU time saved by this premature termination is not significant for the current purpose, and to avoid complications in the dynamic updates described in the following sections.

To each vertex v, two pieces of information are attached: The best-path distance from the source, $\delta(v)$ (here: the accumulated cost to reach v from v_R), and the predecessor vertex in the best-path, $P(v)$. The predecessorships define a directed tree, the *best-path-tree*, and we may follow it from any vertex v back to the source of the tree, v_R, to reconstruct the best path from v_R to v. Upon initialization, the distances are set to $\delta(v_R) = 0$ and $\delta(v) = \infty \, \forall \, v \in \mathcal{V} \backslash \{v_R\}$, while the predecessors

are undefined. All vertices $v \in \mathcal{V}$ are added to a todo-list Q. Subsequently, the *Dijkstra iteration* loop starts which runs until Q is empty. In each iteration, the vertex v with the smallest distance $\delta(v)$ (in the first iteration, this is v_R) is removed from Q. Then, all neighbors u of v are checked. If it is found that the distance to u *via* v is shorter than its current shortest distance, $\delta(u)$, following *correction* is made:

$$\begin{aligned} \delta(u) &:= \delta(v) + c_{uv}, \\ P(u) &:= v. \end{aligned}$$

An illustration of the Dijkstra algorithm is given in Fig. 2.6. The time complexity of the Dijkstra algorithm is $O(|\mathcal{E}|\log|\mathcal{V}|)$ (this is achieved if a Fibonacci heap[3] is used for Q). When applied to Transition Networks, the Dijkstra algorithm is guaranteed to produce a globally optimal best-path-tree[4].

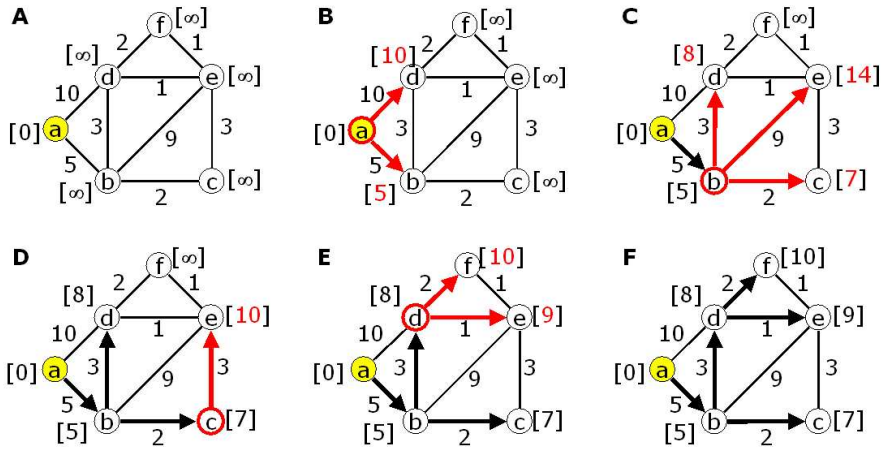


Figure 2.6: Illustration of the Dijkstra algorithm. The best path and the distance from vertex (a) to each other vertex is computed. A) Initialize distance to (a) with 0 and the distances of all other vertices with ∞ (distances in brackets). All vertices are added to a todo-list. B-F) Until the todo-list is empty, the vertex with the smallest distance is removed from it and updated. During the update of vertex v, all neighbors u of v are considered (red arrows). If any neighbor u can be reached from the v by a shorter distance than previously, the distance and predecessor of u is corrected (changes in red). For each vertex, its predecessor is at the root of the arrow pointing to it.

[3] A Fibonacci-Heap is a data structure that can be used as a sorted queue. One can efficiently add elements (in constant time) with a defined priority to the heap in an arbitrary order, while always retrieving the element with the highest priority (in logarithmic time). Here, the elements are the TN vertices and their priority is given by their negative distances.

[4] Transition Networks have non-negative costs. For graphs containing negative weights, Dijkstra fails and other methods have to be used [69, 70].

2.3.2 Dynamical Cost Update

As described in Secs. 2.3.3 and 3.2, we will have situations where the best path needs to be recomputed many times, after changing a single edge cost each time. To avoid unnecessary overhead in the re-computations, the algorithms of Frigioni and Marchetti-Spaccamela [71] are used to dynamically update the shortest path tree with minimal effort. These algorithms distinguish between two cases, the decrease and the increase of the edge cost between (u, v) from its old value c_{uv} to its new value c'_{uv}. Without loss of generality we define $\delta(v) > \delta(u)$ (we choose v to be the vertex with the larger distance).

The cost-decrease ($c'_{uv} < c_{uv}$) only has an effect on the best-path tree if it reduces $\delta(v)$: If the edge (u, v) is not part of the best-path tree, a cost-change in (u, v) might be without effect on the best-path-tree. If, on the other hand, $\delta(v)$ is reduced, then the distances of all vertices in the best-path-subtree with v as a root will also decrease. Furthermore, additional vertices might be added to this best-path subtree. This happens when the changed edge (u, v) allows them to be reached by a new and less expensive path *via* (u, v). The dynamic update is computed by adding v to the todo-list Q and starting the Dijkstra iteration cycle. In contrast to the standard Dijkstra iteration, whenever the distance of a vertex changes, this vertex is also added to Q. In this way, the whole subtree with v as a root and all other affected parts of the graph are updated. An illustrative example is given in Fig. 2.7.

The cost-increase ($c'_{uv} > c_{uv}$) has an effect on the best-path tree exactly if (u, v) is part of the best-path-tree. In this case, the complete subtree with v as a root is affected. In a first step, all vertices of the subtree are marked for being processed by 'coloring them red' [5]. All vertices i at the border of the red region (those which have at least one non-red neighbor) are assigned the minimum possible distance $\delta(i) := \delta(j) + c_{ij}$ and predecessor $P(i) := j$, where j is any non-red neighbor of i. All red vertices are added to Q and the Dijkstra iteration cycle is started normally.

[5] In the original formulation of the algorithm, vertices can also be colored in blue, which means that their predecessor changed, but their distance did not because they can be reached by another best path with equal cost. This does not change the result, but allows to save computation time because best-path sub-trees starting from blue vertices are unchanged. As Transition Networks operate with non-integer weights this case is extremely unlikely and therefore not considered here.

An illustrative example is shown in Fig. 2.8. The proofs and complexity analyses of these algorithms are given in [71].

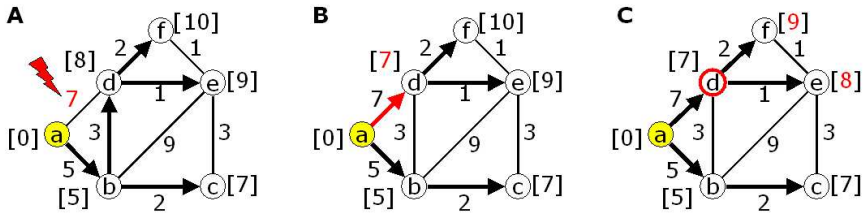


Figure 2.7: Re-computation of the best-path tree when a single cost is decreased. (A) The cost (a-d) is decreased from 10 to 7 (red flash). (B) This cost-change decreases the distance to vertex d ($\delta(a) + c_{ad} < \delta(d)$) , whose predecessor is changed to vertex (a). Vertex d is added to the todo-list. (C) The Dijkstra iteration is started. In each iteration, a neighbor whose distance is decreased is also added to the todo-list. Here, vertices d,f,e are updated until the todo-list is empty and the algorithm terminates.

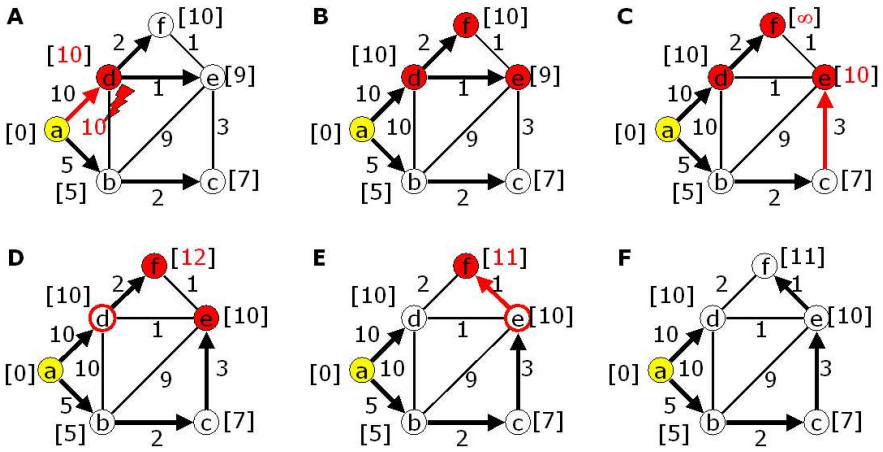


Figure 2.8: Re-computation of the best-path tree when a single cost is increased. (A) The cost of (b-d) is increased from 3 to 10. There is now a new best path from (a) to (d) because $\delta(a) + c_{ad} < \delta(b) + c_{bc}$. The distance to (d) and its predecessor are changed accordingly. (B) All vertices in the best-path subtree with vertex d as a root are marked red ("to be updated") and are added to the todo-list. (C) All vertices at the border of the red region (those which have at least one non-red neighbor, here (d) and (e)) are assigned the nearest non-red neighbor as predecessor and are given an appropriate distance value. (D-E) The Dijkstra iteration cycle is started, each updated vertex is colored white. (F) shows the new best-path tree after the update.

2.3.3 Multiple Best Paths

To obtain an idea of the number of different accessible pathways and their associated structures, it is useful to determine the set of k different pathways, $(P_1, P_2, ..., P_k)$ with costs $(C_1 \leq C_2 \leq ... \leq C_k)$, where P_1 is the path with the lowest cost, C_1, P_2 is the path with the second-lowest cost, C_2, *etc.* This so-called "k best path problem" is well-known in graph theory [72]. To precisely define it, one must define in which way two paths must differ in order to be treated as different. In a transition network, it is clearly not very meaningful to distinguish two pathways which differ only in two low-energy, non-rate-limiting barriers. Therefore, two paths are treated as different only if their rate-limiting steps (*i.e.* their highest-energy edges) do not coincide. The k best paths are determined in k steps: The second best path is found by using the Dijkstra algorithm after "blocking" the edge (u, v) associated with the highest energy barrier in the previously-found best path (by setting its $E_{uv}^{TS} = \infty$). The third best path is found by blocking the highest edges of the best and second best paths, *etc.*

2.4 Energy Ridges

The collection of rate-liming transition states from all different (as defined above) paths from a defined reactant to a defined product belongs to a (D-1)-dimensional transition surface that divides the D-dimensional conformation space into a reactant and a product side. In terms of topography, this transition surface corresponds to an *energy ridge*, as illustrated in Fig. 2.4. On a geographical landscape, it is analogous to a water-shed, *i.e.* the mountain ridge that separates water flows towards distinct oceans. The particular interest of the energy ridge, is that it allows to quickly get a feeling for how degenerate the transition is, *i.e.* how many significantly different paths are likely to be accessible. For instance, if one transition state in the ridge has a significantly lower energy than the other transition states in the ridge, then the transition mechanism is dominated by a well-defined bottle-neck. In contrast, if the ridge contains many different transition states with similar energies, the transition mechanism is not well defined.

In graph-theoretical terms, an energy ridge is a *cut*. The name "cut" stems from

the fact that deletion of its edges dissociates the network into two disconnected subnetworks. Formally, the cut C is a set of M edges $C = \{(u_1, v_1), ..., (u_M, v_M)\}$ with the property that each vertex u_i belongs to one set, U (*e.g.* "reactant side"), each vertex v_i belongs to another set, V (*e.g.* "product side"), and (U, V) partition the set of all vertices (*i.e.*, $U \cup V = \mathcal{V}$ and $U \cap V = \emptyset$).

When the best and all next-best paths each have a dominant (rate-limiting) step, the energy ridge is identical to the cut whose total flux k_{UV} across it is minimal. k_{UV} is given by the sum of all localized rates $k_{u_i v_i}$ in the direction $U \to V$ across edges (u_i, v_i) in the cut:

$$k_{UV} = \sum_{(u_i, v_i) \in C} k_{u_i v_i}, \tag{2.16}$$

where $k_{u_i v_i}$ is the equilibrium rate from Eq. (2.8). By dismissing the constant pre-factor γ/Z, we obtain the normalized total flux, $k_{UV,0}$:

$$k_{UV,0} = \sum_{(u_i, v_i) \in C} \exp\left(-\frac{E_{u_i v_i}}{k_B T}\right). \tag{2.17}$$

Note that the cut that minimizes $k_{UV,0}$ (the *rate-limiting cut*) and the cut associated with the topographic energy ridge are not always identical. For example, consider a case where the topographic ridge is very broad, its cut containing many edges of similar energy, whereas another cut contains only a single edge of slightly lower energy than those of the topographic ridge. Then the cut with the single edge has a lower $k_{UV,0}$ than the cut of the topographic ridge, because the many individual fluxes across the broad topographic ridge add up to a larger total flux. In the current context, however, this theoretical difference is not of importance.

The rate-limiting cut can be found by defining the vector of weights $\mathbf{w} = (w_{\mathcal{E}_1}, ..., w_{\mathcal{E}_{|\mathcal{E}|}})$, where for each edge (u, v) in the network $w_{uv} = \exp(-E_{u_i v_i}/k_B T)$, and minimizing the total weight of the cut using the algorithm of Nagamochi and Ibaraki [73]. However, this algorithm is computationally expensive (scaling as $O(|\mathcal{V}|^3)$ or $O(|\mathcal{V}|^2 + |\mathcal{V}||\mathcal{E}|\log|\mathcal{V}|)$, depending on the implementation). Since the computation

of the cut has to be repeated many times (see Sec. 3.3), we used the topographical energy-ridge cut rather than the rate-limiting cut.

2.4.1 Computation of the Energy Ridge

The topographical energy-ridge cut is determined by an algorithm that can be likened to flooding the energy landscape by stepwise filling up its basins. The ridge that last divides the reactant and product "lakes" before they become connected is the energy ridge. This ridge is similar to the rate-limiting cut, because the Boltzmann weight of a set of edges is most likely dominated by the lowest-energy edge. This is formulated by following proposition:

Proposition: For any two sets of edges $\mathcal{E}_1$ and $\mathcal{E}_2$ in a TN, it holds that:

$$\min\{E_{i\in\mathcal{E}_1}\} < \min\{E_{j\in\mathcal{E}_2}\} \Rightarrow \sum_{i\in\mathcal{E}_1} \exp\left(-\frac{E_i}{k_B T}\right) \gtrapprox \sum_{j\in\mathcal{E}_2} \exp\left(-\frac{E_j}{k_B T}\right)$$

Assuming that this proposition is exact rather than approximate, the following iterative definition for the energy ridge can be given:

1. Select a cut such that its lowest-energy edge is as high as possible, otherwise the cut is arbitrary. Add the lowest-energy ridge to the set ER.
2. Select a cut which has the edges in ER as members and whose second-lowest-energy edge is as high as possible, otherwise the cut is arbitrary. Add the second-lowest-energy ridge to the set ER.
3. Continue this procedure until the cut is fully defined. It now coincides with the energy ridge and its edges are stored in the set ER

To find the ridge according to this definition, the edges which define the energy ridge are identified iteratively, starting from an edge-less network, $\mathcal{G}$, consisting only of the vertices, $\mathcal{V}$. In each iteration, a new edge $e \in \mathcal{E}$ is added to the network in order of increasing edge energy. At each iteration, the topology of $\mathcal{G}$ allows to identify connected subgraphs (*i.e.* sets of vertices in which each vertex has at least one link to another vertex in the set). Each vertex is assigned an identifier that is unique for the connected subgraph it belongs to. The subgraph containing the

reactant vertex is always assigned the identifier i_R while the subgraph containing the product vertex is always assigned i_P. Whenever an edge would be added that connects two vertices with identifiers i_R and i_P, this edge is not added, but marked as part of the energy ridge. The full ridge is determined when all edges have been iterated. An illustrative example for the algorithm is shown in Fig. 2.9 and a formal description is given in the pseudo code below. In the worst case, the algorithm runs at $O(|\mathcal{E}|\log|\mathcal{E}| + |\mathcal{V}|\log|\mathcal{V}|)$. Proofs of the correctness and complexity are given in Appendix A.1.

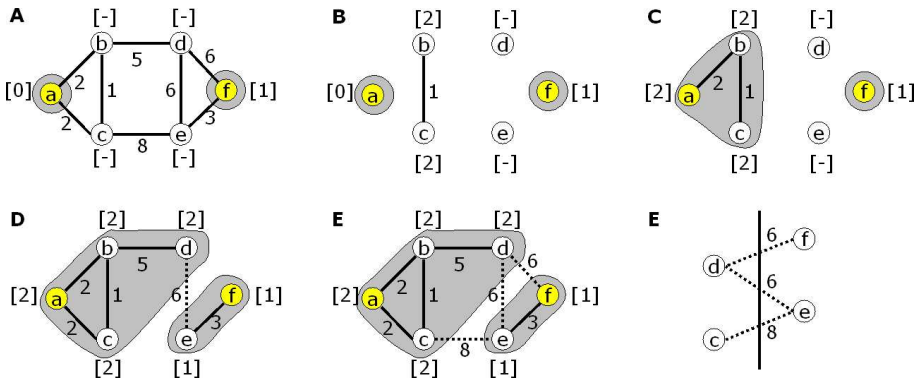


Figure 2.9: Illustration of the computation of energy ridges. (A) shows the full network with transition end-states a and f, which are initially associated with vertex groups 0 and 1, respectively (see brackets). (B-C). All edges are removed and re-included into the network in an order of nondecreasing edge energy. Whenever an edge is placed, the two groups of vertices connected thereby are associated to a common group. (D-E) Edges which would connect the end-state-groups are not placed, but rather marked as members of the energy ridge. (F) shows the set of edges comprising the energy ridge.

2.4.2 Dynamic Edge Energy Update

Similarly as for the best path, there are situations where the energy ridge needs to be recomputed many times, after changing a single edge weight each time (see Sec. 3.3). To avoid repeating the full algorithm 1 each time, a quick update scheme is introduced here.

For this, the list of ordered edges computed in step (1) of algorithm 1 is stored rather than being recomputed each time. Upon the change of a single energy E_{uv} to E'_{uv}, the old weight is removed and the new weight inserted into the sorted

Algorithm 1 Energy Ridge

1) Sort the edges $e = (u, v) \in \mathcal{E}$ by nondecreasing energies E_{uv}.
2) Define an array of vertex group identifiers $F := [-1 ... -1]$ with an entry for each vertex. Assign end-states to groups 0 and 1: $F[v_R] := 0$, $F[v_P] := 1$.
   Set next group identifier: $i_G := 2$.
   Initialize a yet empty set of energy ridge edges $B := \emptyset$.
3) For each $e = (u, v) \in \mathcal{E}$. Define u and v such that the group with identifier $F[u]$ is the larger one; if the groups have equal sizes, define u and v such that $F[u] \geq F[v]$.
   3.1) If ($F[u] = F[v_R]$ and $F[v] = F[v_P]$):
   $ER = ER \cup e$ (e is part of the ridge).
   3.2) Else if $F[u] = -1$ and $F[v] = -1$:
   $F[u] := i_G$, $F[v] := i_G$. Increment i_G.
   (form new vertex group).
   3.3) Else if $F[u] \neq F[v]$:
   For all vertices w with identifier $F[v]$: $F[w] := F[u]$.
   (join two vertex groups)
4) Return ER.

list in the appropriate place. Using sorted heaps, these operations are of order $O(\log|\mathcal{E}|)$.

If the energy is decreased ($E'_{uv} < E_{uv}$), and (u, v) is not part of the ridge, the ridge does not change. If it is part of the ridge, the ridge may change and steps (2)-(4) of algorithm 1 are executed with the new sorted edge list. The energy decrease therefore has a maximum complexity of $O(\log|\mathcal{E}| + |\mathcal{V}|\log|\mathcal{V}|)$ (see Appendix A.1).

If the energy is increased ($E'_{uv} > E_{uv}$), and (u, v) is part of the ridge, the ridge does not change. If (u, v) is not part of the crest and the new energy is not larger than the minimum energy of the ridge ($E'_{uv} \leq E_{\text{pass}}$), the energy ridge does also not change, as this cannot generate a new ridge whose minimum energy is higher than the one of the present ridge. if $E'_{uv} > E_{\text{pass}}$, however, the ridge may change in such a way that the new ridge contains (u, v). This can only be the case if there is no pathway connecting u and v whose maximum edge energy is lower than E_{pass} because the existence of such a pathway would mean that the minimum energy of a cut through (u, v) must also be lower than E_{pass}. The existence of this pathway can be checked with a breadth-first search starting from u which explores only neighbors than can be reached by edges with energies less

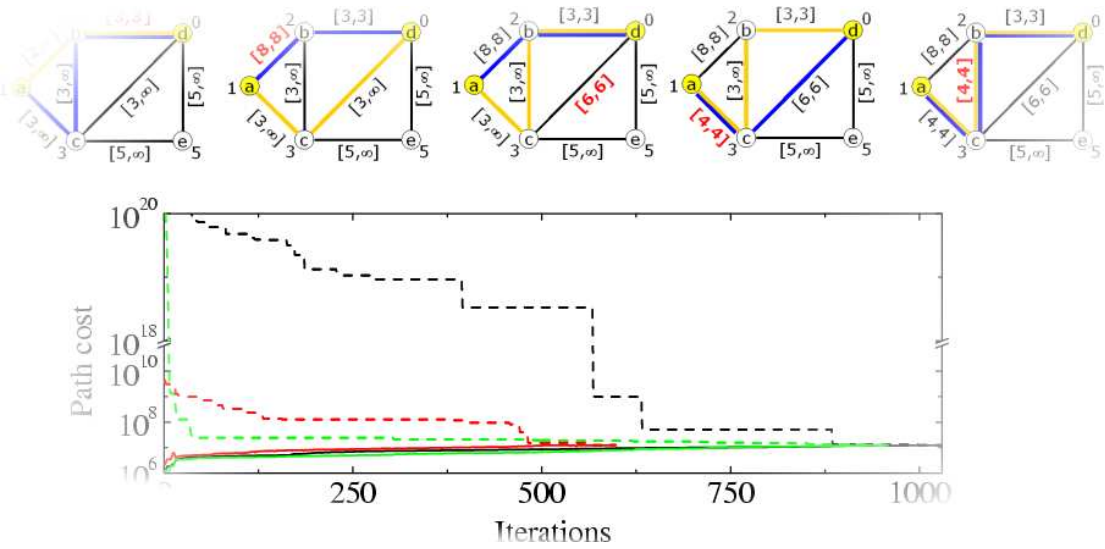
than E_{pass}. The breadth-first search has complexity $O(|\mathcal{E}| + |\mathcal{V}|)$ [74], therefore the maximum complexity of the energy increase is $O(|\mathcal{E}| + \log|\mathcal{E}| + |\mathcal{V}|\log|\mathcal{V}|)$ (see Appendix A.1).

2.5 Conclusion

Global properties of the molecular system, such as best paths and energy ridges can be obtained from a Transition Network for that system. By definition, the construction of the TN does not depend on the height of energy barriers on the energy surface. This circumvents the main problem with molecular dynamics simulations whose sampling time is typically too short to observe events that involve high-barrier passage. However, two other problems must be solved for the construction of TN: 1) The choice of molecular conformations (which yield the static TN vertices) is trivial only for simple molecules, but difficult for complex systems such as proteins. The solution to this problem depends on the class of molecular systems, and a solution specific to proteins will be given later in Chapter 4. 2) The computationally expensive calculations involved in determining energy barriers are too demanding to determine all edge weights for large TN. The solution of this problem is general to all molecular systems and will be covered in the next chapter.

Chapter 3

Efficient Determination of Best Paths and Energy Ridges



The determination of edge energies is computationally very expensive (see Sec. 2.1.1). It is typically not affordable to determine all edge energies of a large static TN. Nevertheless, we want to determine global properties of the network, such as the best path connecting the pair of vertices which corresponds to the end-states of a transition (Sec. 2.3) and the energy ridge separating these vertices (Sec. 2.4). However, the Dijkstra and the energy ridge algorithms introduced above require the knowledge of all edge energies of the static TN. This dilemma is resolved by the algorithms presented in this chapter.

Sec. 3.1 extends the definition of Transition Networks by introducing TN with bounded weights. These bounds express the fact that the TN edge weights (*i.e.* energies or costs) are initially unknown, but are nevertheless guaranteed to be within a certain range of values. When additional knowledge on a certain edge weight becomes available, which allows to narrow the range of possible values for this weight, this is expressed by changing the bounds accordingly. Sec. 3.2 presents an algorithm that allows to compute the best path (or multiple best paths) using weight-bounded TN. This algorithm exploits the fact that in realistic TN only a few edges actually contribute to the determination of the best path while most edges can be ignored solely based on their energy bounds. The algorithm singles out the relevant edges in an iterative manner and determines their energies and costs until the best path is determined. Sec. 3.3 present a very similar algorithm that allows to compute the energy ridge. Sec. 3.4 shows how the performance of both algorithms can be increased by 1) determining the selected energy barriers in several steps rather than in a single step, by 2) introducing uncertainty on the less relevant parts of the best path or energy ridge or by 3) using statistically estimated bounds on the edge energies (and thereby costs). Sec. 3.5 proposes a parallel version of the algorithms introduced here.

3.1 Weight-bounded Transition Networks

Weight bounds express the fact that TN weights are initially unknown, but are nevertheless guaranteed to be within a certain range of values. As shown later (3.4.3) the use of statistical estimates as bounds can also be desirable. The meaning of statistically derived bounds is that the edge weights are expected to reside

within these bounds with a certain probability.

For static TN, we define two edge energy bounds, the lower bound $E_{uv}^{\min} \leq E_{uv}$ and upper bound $E_{uv}^{\max} \geq E_{uv}$. If there is no knowledge available on the edge energies, we use following *a priori* bounds:

$$E_{uv}^{\min} := \max\{E_u, E_v\} \tag{3.1}$$
$$E_{uv}^{\max} := \max\{E_u, E_v\} + M, \tag{3.2}$$

where M is a number that is larger than the anticipated maximum energy barrier[1]. In some cases, it is convenient to refer to relative barriers instead of absolute energies, so we define the edge barrier B_{uv}[2]:

$$B_{uv} := E_{uv} - \max\{E_u, E_v\}, \tag{3.3}$$

such that Eqs. 3.1 and 3.2 translate into

$$B_{uv}^{\min} := 0 \tag{3.4}$$
$$B_{uv}^{\max} := M. \tag{3.5}$$

Formally, we define two static TN, $\mathcal{G}^{\min}$ with the lower bounds as edge energies, and $\mathcal{G}^{\max}$ with the upper bounds as edge energies. From these, a pair of dynamic TN, $\mathcal{G}_{\bar{W}}^{\min}$ and $\mathcal{G}_{\bar{W}}^{\max}$, are derived, which have lower and upper edge costs, $c_{uv}^{\min}$ and $c_{uv}^{\max}$. An edge is said to be *undetermined* if its edge weight is *unknown* (*i.e.* if $E_{uv}^{\min} \neq E_{uv}^{\max}$ or $c_{uv}^{\min} \neq c_{uv}^{\max}$). It is said to be *determined,* if its edge weight is *known* (*i.e.* $E_{uv} = E_{uv}^{\min} = E_{uv}^{\max}$ or $c_{uv} = c_{uv}^{\min} = c_{uv}^{\max}$).

3.2 Efficient Computation of Best Paths

We propose an iterative algorithm to compute the best path in a weight-bounded Transition Network while determining only a small number of edge energies.

[1] M is used instead of ∞ because this avoids some numerical problems

[2] B_{uv} is to be distinguished from the usual definition of an energy barrier, which is the difference between transition state and reactant energies (such as in Sec. 2.2.2). Such a definition would produce two different barriers for each edge ($b_{uv} = E_{uv} - E_u$ and $b_{vu} = E_{uv} - E_v$).

3.2.1 Algorithm

Given a TN with bounded edge energies, the following algorithm iteratively determines the best path through the network by identifying, in each iteration, a critical edge, e_{crit}, whose energy barrier is likely to give the most information on the best path. For this, a hypothetical "optimistic" best path $BP^{\min}$ is determined (see Sec. 2.3) as the best path through the "optimistic" TN $\mathcal{G}^{\min}$ whose yet-uncomputed edge energies are taken to be their lower bounds, $E_{uv}^{\min}$. The critical edge e_{crit} is defined as one with the highest energy along that optimistic best path $BP^{\min}$ [3]. Then, the CPU-intensive step is performed by determining the real energy of e_{crit}. This may lead to a different optimistic best path in the next iteration. This is repeated until all edge energies along the optimistic best path have been computed, giving the truly best path.

To obtain a preliminary estimate of the result, the best path can also be determined in each iteration for the "pessimistic" TN $\mathcal{G}^{\max}$ whose yet-uncomputed edge energies are taken to be their upper bounds, $E_{uv}^{\max}$. This "pessimistic" best path $BP^{\max}$ yields an upper limit to the rate-limiting energy barrier, and also to the cost-value of the true best path. During successive iterations both the optimistic and the pessimistic limits converge to the values of the rate-limiting barrier and cost of the true best path (see Fig. 3.1). Following pseudo code gives a formal representation of the algorithm:

Algorithm 2 Efficient determination of best path

1) Compute two best paths: $BP^{\min}$ on $\mathcal{G}^{\min}$ and $BP^{\max}$ on $\mathcal{G}^{\max}$ between v_R and v_P. Output the maximum energy barriers and the cost values for both paths.
2) If all edge energies along $BP^{\min}$ are determined (*i.e.* $E_e^{\min} = E_e^{\max} \forall e \in BP^{\min}$), RETURN($BP^{\min}$).
3) Choose from $BP^{\min}$ the edge e_{crit} with $E_{e_{crit}}^{\max} = \max\{E_f^{\max} | f \in BP^{\min} \wedge E_f^{\min} < E_f^{\max}\}$ (critical edge with undetermined energy). If e_{crit} exists: determine e_{crit}.
4) GOTO 1.

Proofs for the termination and the correctness of the algorithm are given in Appendix A.2. Fig. 3.2 illustrates the algorithm.

[3] In practice, $E_{uv}^{\max}$ instead of $E_{uv}^{\min}$ is used to identify the edge with the highest energy bound, despite the best-path algorithm is otherwise working with the $E_{uv}^{\min}$. This is of advantage when edge energies are determined in multiple steps (see Sec. 3.4.1). To understand the algorithm,

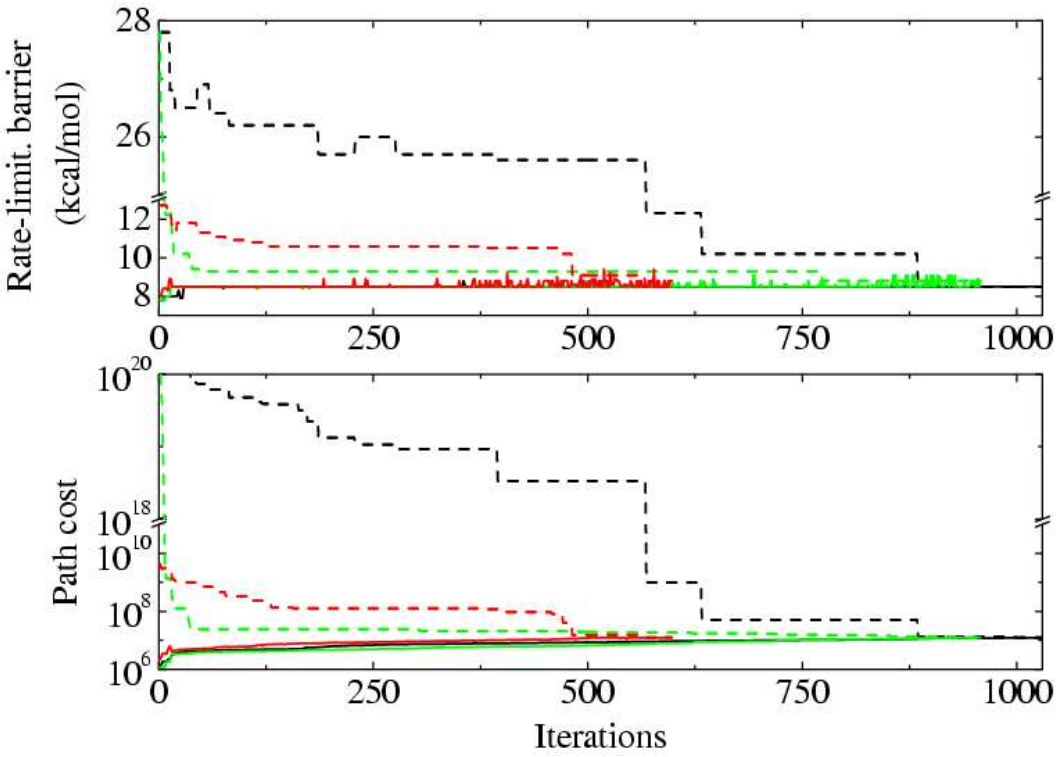


Figure 3.1: Convergence behavior of the best-path, using different settings. The algorithm determines an "optimistic" and a "pessimistic" guess of the best path in each iteration, allowing to derive a lower and an upper bound for the rate-limiting barrier of the true best path (A) and its path-cost (B). These optimistic and pessimistic pairs of values are equal when the true best path is determined. If *a priori* bounds $[0, M]$ are used on the edge energy barriers (black lines), the upper bound for the path-energy and the path-cost is very far from their true values for many iterations but then quickly converges when any full pathway between the transition end-states has been determined on $\mathcal{G}^{\max}$ (*i.e.* when the current pessimistic hypothesis of the best path does not contain any uncomputed barriers with height M). If the values of edge energies are refined in two steps (green lines) rather than one, the initial convergence is much faster, because the first step of refinement is first performed on all edges along the current best path hypothesis before the CPU-intensive full determinations of the edge energies begin. The use of statistical estimates for the edge energy bounds (red lines) also allows the lower and upper estimates for the best-path energy and cost to converge faster, and may significantly speed up the computation.

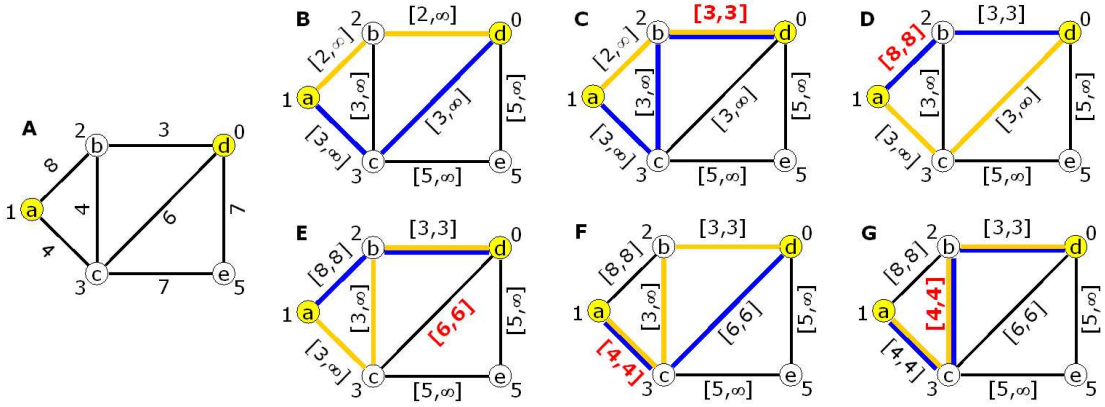


Figure 3.2: Illustration of the algorithm to determine the best path through a weight-bounded TN. A) The TN with its (initially unknown) true edge energies. B) Each edge is given lower and upper bounds for the edge energies (here *a priori* bounds are shown). Best paths are computed between the end-states (a) and (d) on the optimistic and pessimistic TN, giving rise to the yellow and blue paths, respectively. The critical edge (b)-(d) is rate-limiting on the optimistic best path and its true energy is determined. C)-G) The process is iteratively repeated, until all edge energies along the optimistic best path are determined. Both, the optimistic and pessimistic best path coincide with the true best path.

3.2.2 Performance

The CPU time needed for the best path algorithm is dominated by the determination of the edge energies. Therefore, to evaluate the performance of the algorithm, we measure the number of computed edge energies, n_{ec}, necessary to determine the best path.

Theoretical maximum

We first derive an approximate theoretical maximum of n_{ec}. Consider a set of edges, $\mathcal{E}_i$. The inverse Boltzmann weight of this set of edges is most likely dominated by the highest edge energy. We can therefore formulate following proposition:

Proposition: For any two sets of edges $\mathcal{E}_1$ and $\mathcal{E}_2$ in a TN, it holds that:

$$\max\{E_1\} < \max\{E_2\} \Rightarrow \sum_{i\in\mathcal{E}_1} \exp(E_i/k_BT) \lessapprox \sum_{j\in\mathcal{E}_2} \exp(E_j/k_BT)$$

however, this technical detail is not relevant.

Assuming this proposition is exact rather than approximate, then edges (u, v) with $E_{uv}^{\min} > E_{\text{peak}}$ are never refined, where E_{peak} is the highest edge energy of the best path[4]. This gives us following approximate upper bound for n_{ec}:

$$n_{ec} \lessapprox n_{\text{low}} = |\{e \in \mathcal{E} | E_e^{\min} \leq E_{\text{peak}}\}|$$

This upper bound is only approximate because proposition 1b is itself only approximate. However, not as single case with $n_{ec} > n_{\text{low}}$ was observed (see Fig. 3.5).

In most cases, it holds that $n_{ec} < n_{\text{low}}$, as it is not necessary that all edges with $E_{uv}^{\min} \leq E_{\text{peak}}$ are determined. Some of them may lie in regions of the network which are separated from the transition end-states by edges with $E_{uv}^{\min} > E_{\text{peak}}$ and are therefore never considered. A typical example for this is a conformational transition that occurs between two native conformations of a protein, for which transitions in the non-native (*e.g.* unfolded) conformation need not to be considered.

n_{ec} varies strongly depending on the topology of the network, on its edge energies and on the location of the two end-state vertices. The performance of the best-path algorithm was evaluated on an ensemble of random networks and measured in terms of the average number of edges computed to determine the best path(s), $\langle n_{ec} \rangle$.

Evaluating $\langle n_{ec} \rangle$ on Random TN

For all the simulations in this chapter, random networks with $|\mathcal{V}| = 1000$ were generated. The connectivity of these random networks was chosen such as to represent the distribution of edges typical for TN for biomolecules. The distribution of vertex and edge energies is characterized by the order parameter $o \in [0, 1]$. Here, $o = 1/|\mathcal{V}|$ corresponds to a *random noise* network whereas $o = 1$ corresponds to a network over a single harmonic basin with some local random fluctuations (see

[4] *Proof:* In each iteration of Algorithm 2, $BP^{\min}$ is computed using minimum energies $E_{uv}^{\min}$. As an edge determination may only increase, but never decrease $E_{uv}^{\min}$, the next-iteration's $BP^{\min}$ is guaranteed to have an equal or higher inverse Boltzmann weight than the current one. Therefore, the inverse Boltzmann weights of $BP^{\min}$ are nondecreasing until termination. If the above proposition holds exactly, then the maximum edge energies of $BP^{\min}$ are also nondecreasing. When the algorithm terminates, the maximum edge energy of $BP^{\min} \equiv BP^{\max}$ equals, by definition, E_{peak}. This is then the highest energy-edge that was refined. ■

Fig. 3.3 and Appendix B for details)

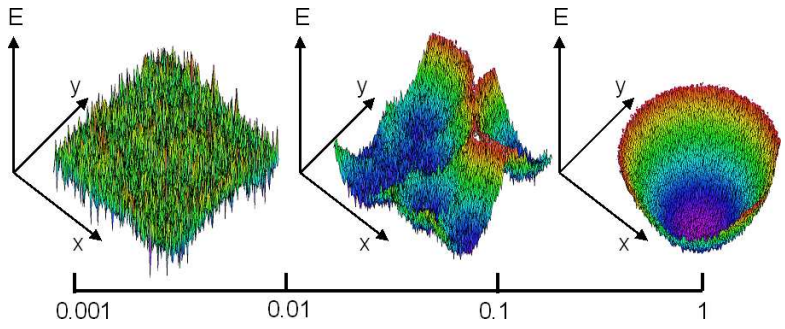


Figure 3.3: Illustration of the effect of the order parameter o on the energy surface underlying of random networks. Here, the energy of a two-dimensional energy surface $E(x, y)$ is shown. A minimum value of $o = 1/|\mathcal{V}|$ (here 0.001) corresponds to a random noise surface, while $o = 1$ represents a harmonic surface with some local roughness.

For each measurement of $\langle n_{ec} \rangle$, 50 random networks were generated and for each of these networks, 50 pairs of end-states were randomly chosen as described in Appendix B and the best paths between them were determined. The *a priori* barrier bounds $B_{uv}^{\min} = 0$ and $B_{uv}^{\max} = 5$ kcal/mol were used. The edge energies were determined in a single step, *i.e.* a determination of (u, v) resulted in setting $B_{uv}^{\min} := B_{uv}^{\max} := B_{uv}$.

We first tested how the form of the underlying energy surface influences the performance of the best-path algorithm, by computing $\langle n_{ec} \rangle$ for random networks with different values for the order parameter o. The results are shown in Fig. 3.4. For random noise networks, $\langle n_{ec} \rangle$ has a maximum value, while for networks with some local structure, $\langle n_{ec} \rangle$ decreases significantly towards a constant value ($o > 0.01$). This result is expected, as random noise networks contain many more pathways of similar energies than TN with more order. The random noise network is therefore a worst-case scenario. It is used as a model for all computations of $\langle n_{ec} \rangle$ in the present chapter, unless stated otherwise.

Fig. 3.5 shows a plot of n_{ec} depending on n_{low} for a total of 10000 best-path computations on random noise networks. Most values of n_{ec} are about 2 orders of magnitude below $n_{ec} = n_{\text{low}}$. The average number of computed edges, $\langle n_{ec} \rangle$ increases linearly with n_{low} for large values of n_{low}.

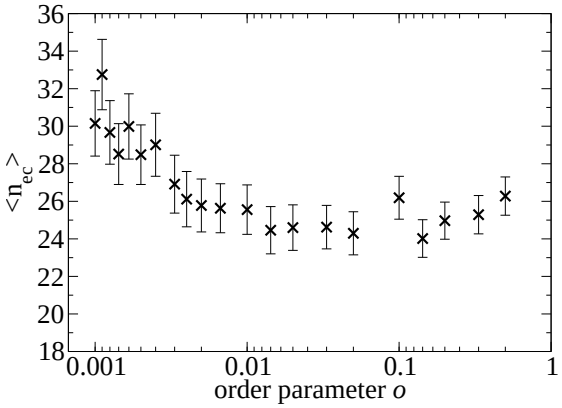


Figure 3.4: Average number of computed edges depending on the amount of order on the energy surface. o=0.5 represents two large harmonic wells with some superimposed noise, while o=0.001 represents the random noise network (see also Fig. 3.3).

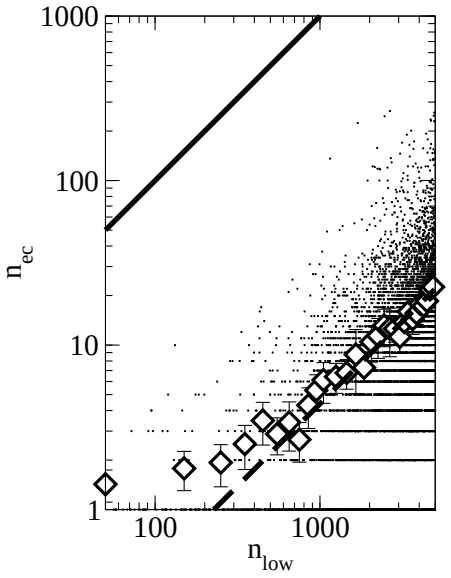


Figure 3.5: The number of computed edges, n_{ec}, for best paths on random networks correlated with the number of low-energy edges, n_{low} (see text). The black dots show all value-pairs that have been generated by simulations on random noise networks. All data points are below the approximate theoretical maximum of $n_{ec} = n_{\text{low}}$ (solid line). The average number of computed edges, $\langle n_{ec} \rangle$ (diamonds) approaches $\langle n_{ec} \rangle = 0.0044\, n_{\text{low}}$(dashed line).

3.2.3 Multiple Best Paths

The best-path algorithm 2 can directly be used to compute multiple, k-best paths, when the protocol described in Sec. 2.3.3 is followed. Typically, there is an overlap between the sets of edges which need to be determined to compute the k-best paths individually. Therefore, the number of edges required to compute k best paths, $\langle n_{ec,k} \rangle$, is expected to be less than the theoretical maximum $k \cdot \langle n_{ec,1} \rangle$, where $n_{ec,1}$ is the number of edges required to compute the best path. This is indeed visible in Fig. 3.6, which shows $\langle n_{ec,k} \rangle$ that has been computed for values of $k \leq 16$. $\langle n_{ec,k} \rangle$ increases linearly with k for the numbers of $\langle n_{ec,k} \rangle$ observed here[5], approaching the function $\langle n_{ec,k} \rangle = 0.78\, k \langle n_{ec,1} \rangle$.

[5] $\langle n_{ec,k} \rangle$ is of course bounded from above by $|\mathcal{E}|$, and can therefore not continue to rise linearly, but the present simulations don't come close to this value

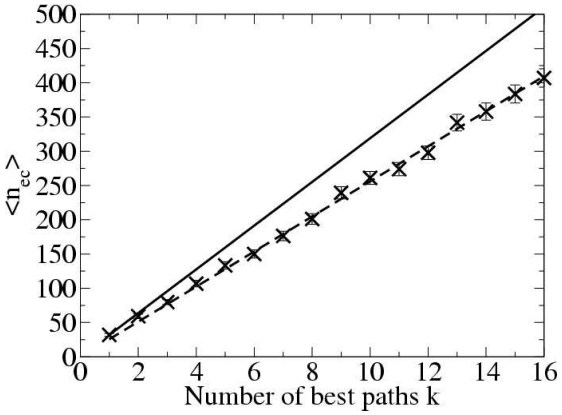


Figure 3.6: Dependency of the average number of computed edges for k-best paths on random networks, $\langle n_{ec,k} \rangle$, on the number k of best paths that were computed. $\langle n_{ec,k} \rangle$ increases slower than the theoretical maximum $\langle n_{ec,k} \rangle = k \langle n_{ec,1} \rangle$(solid line), approaching the function $\langle n_{ec,k} \rangle = 0.78\, k \langle n_{ec,1} \rangle$

3.3 Energy Ridges

Given the best path algorithm for weight-bounded TN, an algorithm for computing the energy ridge that proceeds along the same lines is easily formulated.

3.3.1 Algorithm

As defined in Sec. 2.4, the energy ridge is the set of edges that describes the rate-limiting dividing surface between reactant and product. The algorithm that computes the energy ridge while determining only a limited number of edge barriers uses a strategy similar to the one used to find the best-path (see Sec. 3.2.1). The energy ridge $ER^{\max}$ is computed (see Sec. 2.4.1) on a TN $\mathcal{G}^{\max}$ whose yet uncomputed edge energies are set to the upper edge energy bounds $E_{uv}^{\max}$. Then, the critical edge e_{crit}, here: the lowest-energy edge on this hypothetical energy ridge is determined. This process is repeated in successive iterations. The true energy ridge has been found when all its edge barriers have been determined.

As for the best-path computation, a preliminary estimate of the result can be obtained by computing the energy ridge in each iteration also for the "optimistic" TN $\mathcal{G}^{\min}$ whose yet-uncomputed edge energies are taken to be their lower bounds, $E_{uv}^{\min}$. This energy ridge $ER^{\min}$ yields a lower limit to the true energy ridge's minimum barrier, E_{pass}. During successive iterations these limits converge to the values of the rate-limiting barrier and cost of the true best path (see Fig. 3.1). Following pseudo code gives a formal representation for the algorithm. The proof is analogous to the proof of Algorithm 2 given above.

Algorithm 3 Efficient determination of the energy ridge

1) Compute two energy ridges: $ER^{\min}$ on $\mathcal{G}^{\min}$ and $ER^{\max}$ on $\mathcal{G}^{\max}$ separating v_R and v_P. Output the minimum energy barriers and the weight values for both ridges.
2) If all edge energies along $ER^{\max}$ are determined (*i.e.* $E_e^{\min} = E_e^{\max} \; \forall \, e \in ER^{\max}$), RETURN($ER^{\max}$).
3) Choose from $ER^{\max}$ the edge e_{crit} with $E_{e_{crit}}^{\max} = \min\{E_f^{\max} | f \in ER^{\max} \wedge E_f^{\min} < E_f^{\max}\}$ (critical edge with undetermined energy). If e_{crit} exists: determine e_{crit}.
4) GOTO 1.

3.3.2 Performance

According to an argumentation analogous to the one given in Sec. 3.2.2, we obtain an approximate upper bound for n_{ec} when computing energy ridges:

$$n_{ec} \lessapprox n_{\text{hi}} = |\{e \in \mathcal{E} | E_e^{\max} \geq E_{\text{pass}}\}|.$$

Here, E_{pass} is the minimum energy edge on the ridge (which is typically[6] equivalent to the highest-energy edge of the lowest-energy pathway) and n_{hi} are the number of edges in the network whose upper energy bounds are higher than E_{pass}. As above, this upper bound is only approximate, but no single case with $n_{ec} > n_{\text{hi}}$ was observed.

As for the best path algorithm, we tested how the average number of computed edges, $\langle n_{ec} \rangle$, required to compute the energy ridge depends on the form of the energy landscape underlying the transition network. $\langle n_{ec} \rangle$ was computed using random networks with a different amount of order in the underlying energy surface, o, between $o = 0.001$ (random noise network) and $o = 0.5$ (two harmonic basins with added noise). Fig. 3.7 shows that there is a nearly linear decrease of $\langle n_{ec} \rangle$ with increasing amount of order on the energy surface. As for the best path calculations, the random noise network is the worst-case scenario also for the computation of the energy ridge. It is used as a model for all computations of $\langle n_{ec} \rangle$ for energy ridges in this chapter, unless states otherwise.

Fig. 3.8 shows a plot of n_{ec} depending on n_{hi} for a total of 10000 energy-ridge

[6] This equivalence is not strict as the best path is determined by all barriers along the path (Eq. 2.15) rather than by the rate-limiting barrier alone. For the understanding of the present section, however, this possible difference is not critical.

computations on random noise networks. For small values of n_{hi} (*i.e.* for small networks or very high ridges), n_{hi} is a good upper bound for n_{ec}. For larger values of n_{hi}, n_{ec} is considerably lower and $\langle n_{ec} \rangle$ increases with the square root of n_{hi}.

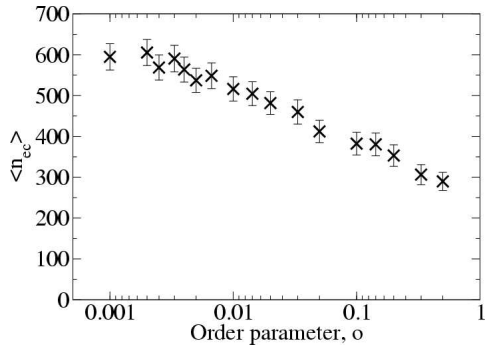


Figure 3.7: Average number of computed edges, $\langle n_{ec} \rangle$ depending on the amount of order of the energy surface, measured by the order parameter o (see Fig. 3.3). A value of o =0.5 represents two large harmonic wells with some superimposed noise, while a value of o=0.001 represents the random noise network.

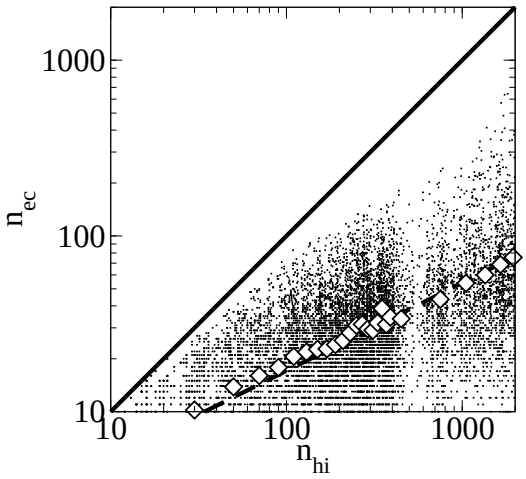


Figure 3.8: The number of computed edges, n_{ec}, for energy ridges on random networks depending on the number of high-energy edges, n_{hi}. The black dots show all value-pairs that have been generated during the simulation. All data points are below the approximate theoretical maximum of $n_{ec} = n_{\mathrm{hi}}$ (solid line). The average number of computed edges, $\langle n_{ec} \rangle$, approached the function $\langle n_{ec} \rangle = 1.7\sqrt{n_{\mathrm{hi}}}$(dashed line).

3.4 Increasing Performance

3.4.1 Stepwise Determination of Edge Energies

In the previous sections we always assumed that edges are determined in a single step. That is, in each iteration of the algorithm the critical edge e_{crit} was chosen, its edge energy $E_{e_{\mathrm{crit}}}$ was determined and its bounds were changed from their initial values $[E^{\min}_{e_{\mathrm{crit}}}, E^{\max}_{e_{\mathrm{crit}}}]$ to $[E_{e_{\mathrm{crit}}}, E_{e_{\mathrm{crit}}}]$. The determination of the edge energy $E_{e_{\mathrm{crit}}}$ is usually performed by an iterative method, which achieves the convergence of the edge energy to the result $E_{e_{\mathrm{crit}}}$ through optimization or sampling (see Secs 2.1.1 and 2.1.3). These methods generally approach the result quickly in the beginning and slower at the end of the computation. Therefore, a considerable

improvement of the edge energy bounds is available at a cheap price by running the determination method only for a few steps and incorporating the preliminary result. Formally, we replace the single-step determination algorithm by a refinement algorithm $\mathsf{refine}(\mathbf{x}_u,\mathbf{x}_v)$ which delivers some refined bounds $E^{\min}_{uv,i+1} \geq E^{\min}_{uv,i}$ and $E^{\max}_{uv,i+1} \leq E^{\max}_{uv,i}$. To assure the termination of the best-path and energy ridge algorithms, we require that for a pair $(\mathbf{x}_u,\mathbf{x}_v)$, at most a finite number of n_R calls to $\mathsf{refine}(\mathbf{x}_u,\mathbf{x}_v)$ are necessary to obtain equal bounds and therefore a determined edge energy: $E_{uv} = E^{\min}_{uv} = E^{\max}_{uv}$. To simulate this approach, we used a two-step refinement algorithm whose first step changed the bounds $[E^{\min}_{uv}, E^{\max}_{uv}]$ into $[E^{\min}_{uv}, E_{uv} + 0.5\,\mathrm{kcal/mol}]$, and whose second step delivered the determined energy $[E_{uv}, E_{uv}]$. Such a multi-step determination approach does not guarantee to find the best path or the energy ridge faster (*i.e.* to reduce n_{ec}). However, it remarkably improves the lower and upper bounds for the energy barrier and the best path cost during the runtime of the algorithm, as shown in Fig. 3.1 (green line). While this is actually an improvement of perceived performance (the user gets a good estimation of the result at an earlier time), a multi-step determination approach can also improve real performance (*i.e.* a reduction of n_{ec}) when used in combination with the method of partial computation introduced below.

3.4.2 Partial Computation of Best Paths and Energy Ridges

The computation time can be significantly reduced if it is not required to compute the best path or the energy ridge in their full detail. We first introduce a strategy for reducing the computing time by introducing uncertainty on the low-energy regions of the best path. One is often not interested in the details of how the best path travels in the low-energy regions, since the highest-energy edges along the whole path are rate-determining. Computation time can thus be saved if only the high-energy edges of the best path, *i.e.*, those with energies within a range ΔE_{sure} of the highest energy along the path, E_{peak}, are requested to be correct (see Fig. 3.9A). To achieve this, Algorithm 2 is extended such that the already-computed edge barriers along the current best path, for which $E_{uv} < E_{\mathrm{peak}} - \Delta E_{\mathrm{sure}}$, are reset as if they were barrier-less transitions (*i.e.* $B_{uv} := 0$). This prevents the next Dijkstra computation from finding different best paths that would circumvent the

low-energy barriers of the current best path.

Fig. 3.10A shows that the use of a ΔE_{sure} can considerably reduce $\langle n_{ec} \rangle$. The amount of reduction, however, depends on the amount of order on the energy surface. For random noise networks $\langle n_{ec} \rangle$ is only reduced by a small fraction when using $\Delta E_{\text{sure}} = 0$ instead of $\Delta E_{\text{sure}} = \infty$, but the reduction amounts 50% for networks with $o = 0.2$. The reason for this difference is that energy surfaces with a significant amount of underlying order, the edge energy bounds already give a good estimate of where the highest edges of the best paths may be located. In contrast, for random noise networks, nearly all edges are candidates for these highest edges, so that the reduction of $\langle n_{ec} \rangle$ compared with a full computation of the best path is small.

As shown in Fig. 3.10B, the reduction of $\langle n_{ec} \rangle$ is stronger, when edge energies are refined in two steps rather than in a single step (see Sec. 3.4.1). In this case, the percentage of $\langle n_{ec} \rangle$ that saved by using $\Delta E_{\text{sure}} = 0$ instead of $\Delta E_{\text{sure}} = \infty$ is approximately doubled. The reason for this is that for many edges which are below $E_{\text{peak}} - \Delta E_{\text{sure}}$, a partial refinement is already sufficient to obtain an upper barrier bound of $B_{uv}^{\max} < E_{\text{peak}} - \Delta E_{\text{sure}}$, after which the edge energy is no longer improved. Since $\langle n_{ec} \rangle$ counts the number of full edge refinements, partial refinements do contribute.

The computation of the energy ridge can be accelerated if only the lowest-energy barriers along the energy ridge are requested, which is often sufficient as higher-

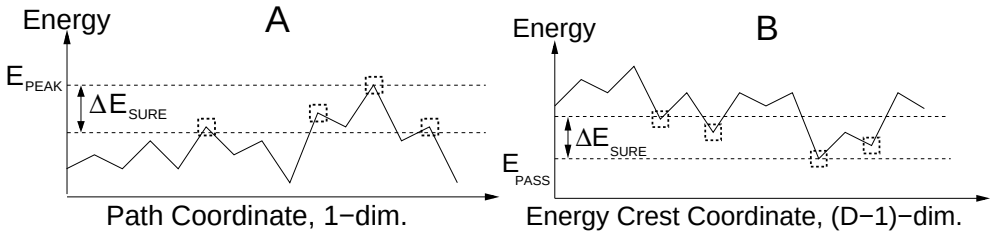


Figure 3.9: Schematic representation of the concept of partial determination of best paths and energy ridges. A) Profile of vertex and edge energies along a pathway through the network. Given ΔE_{sure}, the edges in the range $[E_{\text{peak}} - \Delta E_{\text{sure}}, E_{\text{peak}}]$ are guaranteed to belong to the true best path (indicated by squares). B) Profile of the energy ridge separating two regions of the energy surface. Only edges in the range $[E_{\text{pass}}, E_{\text{pass}} + \Delta E_{\text{sure}}]$ are guaranteed to belong to the true energy ridge.

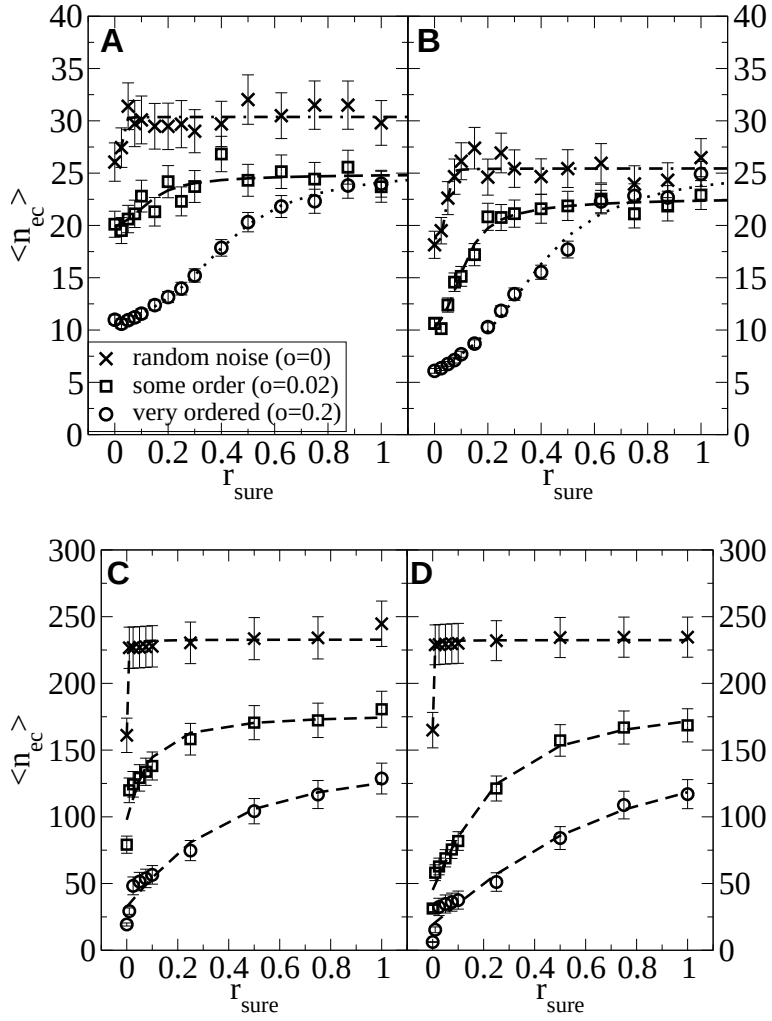


Figure 3.10: Dependency of the average number of computed edges, $\langle n_{ec} \rangle$, for best paths and energy ridges on random networks, on the fraction of the best path or energy ridge that has been determined. A) and B): All path edges with energies in a range of $r_{\text{sure}}(E_{\text{peak}} - E_{\text{min}})$ below the rate-limiting barrier of the path, E_{peak},are guaranteed to belong to the true best path (E_{min} is the minimum vertex energy in the network). A): Edge energies were determined in a single step, B): Edge energies were determined in two steps, after the first step the upper bound was 0.5 above the true edge energy. Using $\Delta E_{\text{sure}} < \infty$ is more beneficial for well-ordered energy surface (bullets, squares) than for random noise networks (crosses). A two-step refinement enhances the savings at low values of ΔE_{sure}. C) and D): All energy ridge edges with energies in a range of $r_{\text{sure}}(E_{\text{max}} - E_{\text{pass}})$ above the ridge's minimum height, E_{pass} are guaranteed to belong to the true energy ridge (E_{max} is the maximum vertex energy in the network). The same edge energy determination procedure was used as in A) and B), respectively.

energy barriers are less likely to be populated. One may only be interested in energy barriers of the energy ridge that are up to ΔE_{sure} above the energy of the lowest energy in the ridge, E_{pass} (see Fig. 3.9B). To achieve this, already computed edge barriers of the current energy crest for which $E_{uv} > E_{\text{low}} + \Delta E_{\text{sure}}$ are reset to $E_{uv} = \infty$. This fools the energy-ridge finding algorithm so that it leaves these high-energy barriers in the ridge. To avoid computing barriers whose energies must be above the energy region of interest (*i.e.*, which have $E_{uv}^{\min} > E_{\text{pass}} + \Delta E_{\text{sure}}$, where E_{pass} is taken from the current guess of the ridge), their energies are also set to $E_{uv} = \infty$, thereby excluding them from being determined at all. The values of $\langle n_{ec} \rangle$ for the partial determination of the energy ridge are plotted in Fig. 3.10C and D. For random noise networks, there is a reduction of $\langle n_{ec} \rangle$ by about 30% if only the rate-limiting step ($\Delta E_{\text{sure}} = 0$) is computed compared to $\Delta E_{\text{sure}} = \infty$. For TN energy surfaces with a higher degree of order, the reduction can be drastic, amounting more than 90%. Again, a two-step refinement procedure is superior to a one-step refinement, but the advantage is less expressed than for the best-path computation.

3.4.3 Statistical Edge Energy Estimates

The computing time can be drastically reduced, at the expense of possibly failing to identify the true best path or energy ridge, if the *a priori* bounds for the edge barriers $B_{uv}^{\min} = 0$ and $B_{uv}^{\max} = M$ are replaced by statistical estimates.

For the best-path calculation, the problem exists that the lower energy bound is not necessarily correct and might overestimate some barriers. That is, edges which are not included in the resulting best path and have been rejected based on their lower estimate $E_{uv}^{\min}$ might in fact have a true edge energy $E_{uv} < E_{uv}^{\min}$. The highest barrier of the truly best path may be lower by up to the maximal difference found between any estimated lower barrier bound and its *a priori* bound, *i.e.*, the error on the rate-limiting barrier is less than:

$$\max\{B_{uv}^{\min}\}_{\text{all pairs}\,(u,v)} \tag{3.6}$$

Estimated energy bounds may also be used to speed up the computation of energy ridges. Here, the true energy ridge might be missed because the upper barrier

estimate may still underestimate some barriers, which would otherwise be included in the energy ridge. Unfortunately, no *a priori* upper bound lower than M can be specified, such that a maximum error as in Eq. 3.6 cannot be specified.

Fig. 3.11 shows the dependence of $\langle n_{ec} \rangle$ on the confidence ratio c of the lower bound estimate for both best paths and energy ridges. c is defined as follows: For each random network, the lower estimate $B^{\min}$ was set such that $c|\mathcal{E}|$ edge energies were greater than $B^{\min}$, and likewise the upper estimate $B^{\max}$ was set such that $c|\mathcal{E}|$ edge energies were smaller than $B^{\max}$ (*i.e.* c is the fraction of correctly estimated bounds). For the best-path computation, even for $c = 1$, $\langle n_{ec} \rangle \approx 15$ is only about half compared to the value of $\langle n_{ec} \rangle \approx 30$ with *a priori* bounds. (see Fig. 3.4). In general, $c = 1$ does not guarantee that $B^{\min}$ are lower bounds for the barrier. This is because usually only a limited number of barrier energies are known when the statistics are set up such that $c = 1$ only means that $B^{\min}$ and $B^{\max}$ are true bounds for all *observed* barriers. However, the result shows that a statistical estimate that involves only little potential error, can save a considerable amount of CPU time.

Smaller values of c can further reduce $\langle n_{ec} \rangle$ for the best-path calculation by a factor of three. For energy ridges, the relative further reduction of $\langle n_{ec} \rangle$ is not significant.

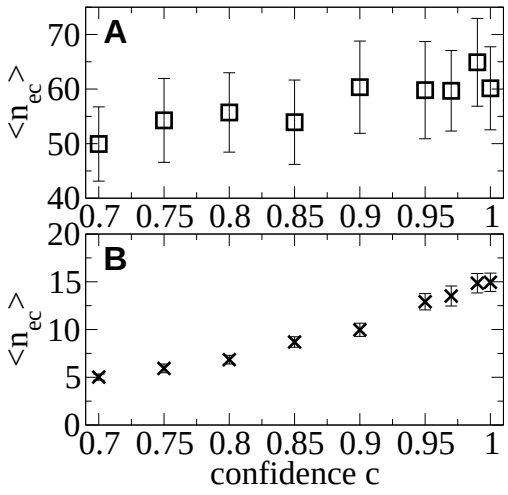


Figure 3.11: Dependency of the average number of computed edges, $\langle n_{ec} \rangle$, for best paths (B) and energy ridges (A) on random networks, on the confidence of statistically-estimated edge energy bounds. For best paths (crosses), even a confidence-value of $c = 1$ (energy bounds are true bounds for all observed network energies), allows to save a factor of two compared to the use of hard edge energy bounds (compare Fig. 3.4).

3.5 Parallelization

Algorithms 2 and 3 can be executed in parallel if each computing node runs one instance of a modified version of the algorithms while the communication is realized by accessing the same database of edge energies. The individual instances need some minimal synchronization to avoid that two instances compute the same edge energy concurrently. For this, it is necessary that an edge can be flagged in the database as being currently computed. If one of the instances determines a flagged edge as critical edge e_{crit}, this instance assigns a hypothetical edge energy $E^{\text{avg}}_{e_{crit}}$ for it, which might be predefined by the user, by a statistical estimate (see Sec. 3.4.3), or simply by $E^{\text{avg}}_{e_{crit}} = \frac{1}{2}(E^{\min}_{e_{crit}} + E^{\max}_{e_{crit}})$. The edge e_{crit} is added to a list of *forged* edges and the algorithm continues to the next iteration, determining another critical edge. In each iteration, the list of forged edges is checked. If the refinement flag for any of these edges has meanwhile been removed, that edge is removed from the list of forged edges and its energy bounds are reset to the current database values. As the presence of forged edges in the network may produce wrong best paths and energy ridges, the parallel implementation of these algorithms requires that the list of forged edges is empty, before it can terminate successfully. As the parallel algorithms for the best path and that for the energy ridge are analogous, only the parallel best path algorithm is given here.

Algorithm 4 Parallel Best-path order

1) Be $F := \varnothing$ a list of "forged" edges.
2) For each member $f \in F$ not flagged as being refined:
   reset $E^{\min}_f$ and $E^{\max}_f$ to their database values, $F := F \backslash \{f\}$.
3) Compute two best paths: $BP^{\min}$ on $\mathcal{G}^{\min}$ and $BP^{\max}$ on $\mathcal{G}^{\max}$ between v_R and v_P. Output the maximum energy barriers and the cost values for both paths.
4) If all edge energies along $BP^{\min}$ are determined (*i.e.* $E^{\min}_e = E^{\max}_e \; \forall \, e \in BP^{\min}$) and $F = \varnothing$: RETURN($BP^{\min}$).
5) Choose from $BP^{\min}$ the edge e_{crit} with $E^{\max}_{e_{crit}} = \max\{E^{\max}_g | g \in BP^{\min} \wedge E^{\min}_g < E^{\max}_g\}$ (critical edge with undetermined energy). If no such edge exists, GOTO 2.
6) If e_{crit} is flagged as being currently computed, assign a hypothetical edge energy to it: $E^{\min}_{e_{crit}} = E^{\max}_{e_{crit}} = E^{\text{avg}}_{e_{crit}}$. Add this edge to the list of "forged" edges :$F := F\{e_{crit}\}$.
   Else flag e_{crit} as being currently computed, determine it and then remove the flag thereafter. GOTO 2.

As the forged edges may be set to wrong values temporarily, and these values determine which edges are determined next, the local instances of the parallel algorithm may explore regions of the network which are not relevant to determine the best path or the energy ridge. Because of these unnecessary computations, there is some loss of efficiency with increasing amount of parallelization. To quantify this effect, we have simulated the computation of best paths and energy ridges on random networks in parallel on a number of p virtual processors. In this simulation, all refinement jobs were assumed to have equal run-times. The total number of edges computed in one such simulation is $\langle n_{ec,p} \rangle$, giving rise to an average runtime per processor of $\langle n_{ec,p} \rangle / p$. The normalized runtime compared to the single-processor process is $\langle n_{ec,p} \rangle / (\langle n_{ec,1} \rangle p)$. Thus, the speed-up is defined as

$$\text{speed} - \text{up} := \frac{\langle n_{ec,1} \rangle p}{\langle n_{ec,p} \rangle}$$

and the efficiency as:

$$\text{efficiency} := \frac{\langle n_{ec,1} \rangle}{\langle n_{ec,p} \rangle}.$$

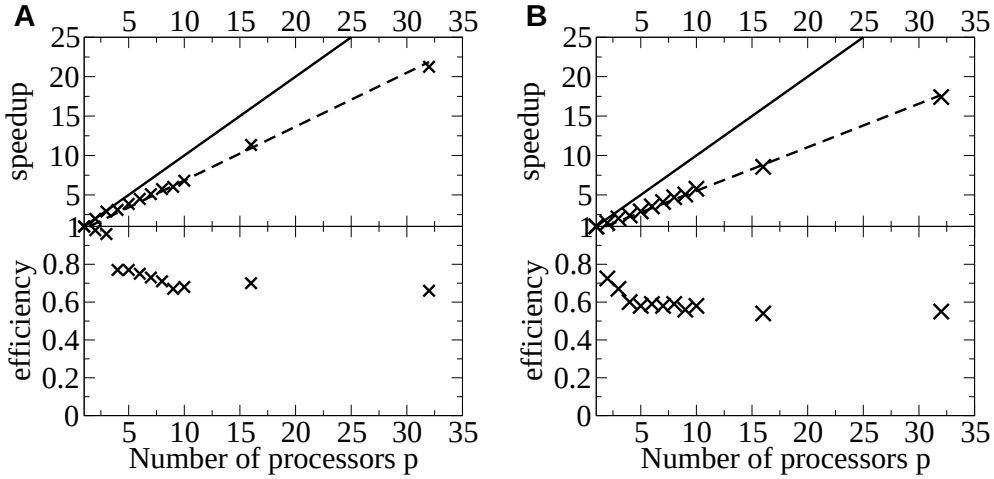


Figure 3.12: Dependency of the parallelization efficiency and speed-up for best paths and energy ridges on random networks, on the number of parallel processors, p, the job was distributed on. A) the speed-up for parallel best paths increases approximately as $0.7p$ (crosses), the solid black line shows the theoretical maximum (speed-up $= p$). The efficiency is nearly 1 for up to three processors and drops to a value of 0.7 for larger p's. B) For energy ridges, the speedup increases approximately as $0.55p$, the efficiency drops to 0.55 at 16 processors.

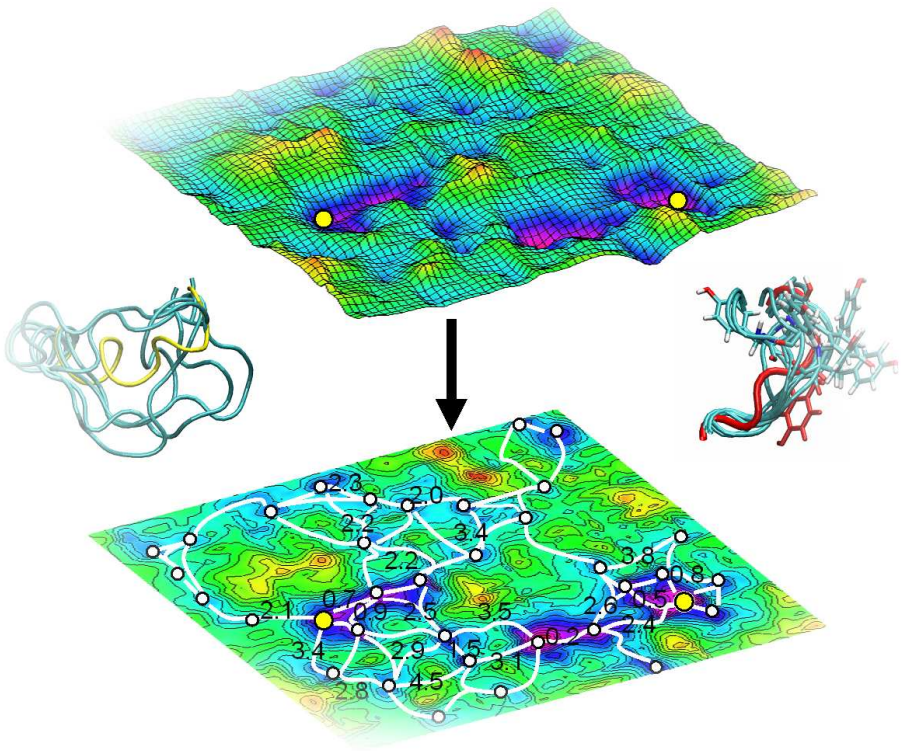
The results are shown in Fig. 3.12. For a large number of processors, the speed-up increases linearly with the number of processors and the efficiency is therefore constant. Computing best paths on up to three processors is practically lossless, the efficiency being near 1. For a larger number of processors, we have a $\text{speed}-\text{up} \approx 0.7p$ (efficiency of 0.7) for the parallel best path and $\text{speed}-\text{up} \approx 0.55p$ (efficiency of 0.55) for the parallel energy ridge.

3.6 Conclusion

The algorithms presented here allow to efficiently determine the best path and the energy ridge. The number of edges that need to be determined is generally several orders of magnitude below the total number of edges in the network. Using the very efficient parallel version presented here, this allows to compute best paths and energy ridges for large Transition Networks of complex systems such as proteins in several days or weeks on a small PC cluster. Here, we conclude the system-independent part of the work and focus on TN for proteins. The following chapter addresses the problem of how to generate appropriate conformational samples that yield the TN vertices for complex conformational changes in proteins.

Chapter 4

Hierarchical Sampling Method for Complex Rearrangements in Proteins



For all but the simplest systems, the determination of the TN vertices $v \in \mathcal{V}$ is nontrivial. Ideally, the set of vertices $\mathcal{V}$ should represent the full range of system configurations that are significant to the process under observation, irrespective of the system size and the location of these configurations. In general, the sampling problem (see Introduction) limits the generation of such a set of vertices for systems with many degrees of freedom. Despite there is no general recipe to overcome the sampling problem without making severe approximations, the problem can be alleviated if one concentrates on finding a solution for a restricted class of systems and processes, exploiting the particular structure of this class. The present chapter proposes an efficient sampling method which is specific to the system class of proteins and to the process class of (possibly complex) conformational changes. We illustrate the method by generating a set of TN vertices for the conformational switch in the Ras p21 protein.

Available *importance sampling* methods, such as molecular dynamics or Metropolis Monte Carlo, successfully generate Boltzmann distributions, in which low-energy conformations are strongly preferred over higher-energy conformations. For this reason, they are inefficient in sampling the full conformational space if this requires crossing high energy barriers. To avoid this problem, the present sampling procedure first generates conformers *uniformly distributed* over a conformational (sub)space and subsequently retains only the low-energy conformers. To obtain a sufficient sampling density, the dimensionality of the sampling problem must be reduced. Therefore, the degrees of freedom which only contribute by flexible deformations are interpolated (Sec. 4.2) while the sampling subspace includes those degrees of freedom that undergo significant changes to facilitate the conformational change to be studied (Sec. 4.3). Out of the generated conformers, only collision-free structures are retained (Sec. 4.4) and minimized (Sec. 4.5). Subsequently, the number of conformers in the low-energy regions is increased until a desired density is obtained (Sec. 4.6). Figs. 4.2 and 4.1 show schematic overviews of the steps involved in the generation of TN vertices.

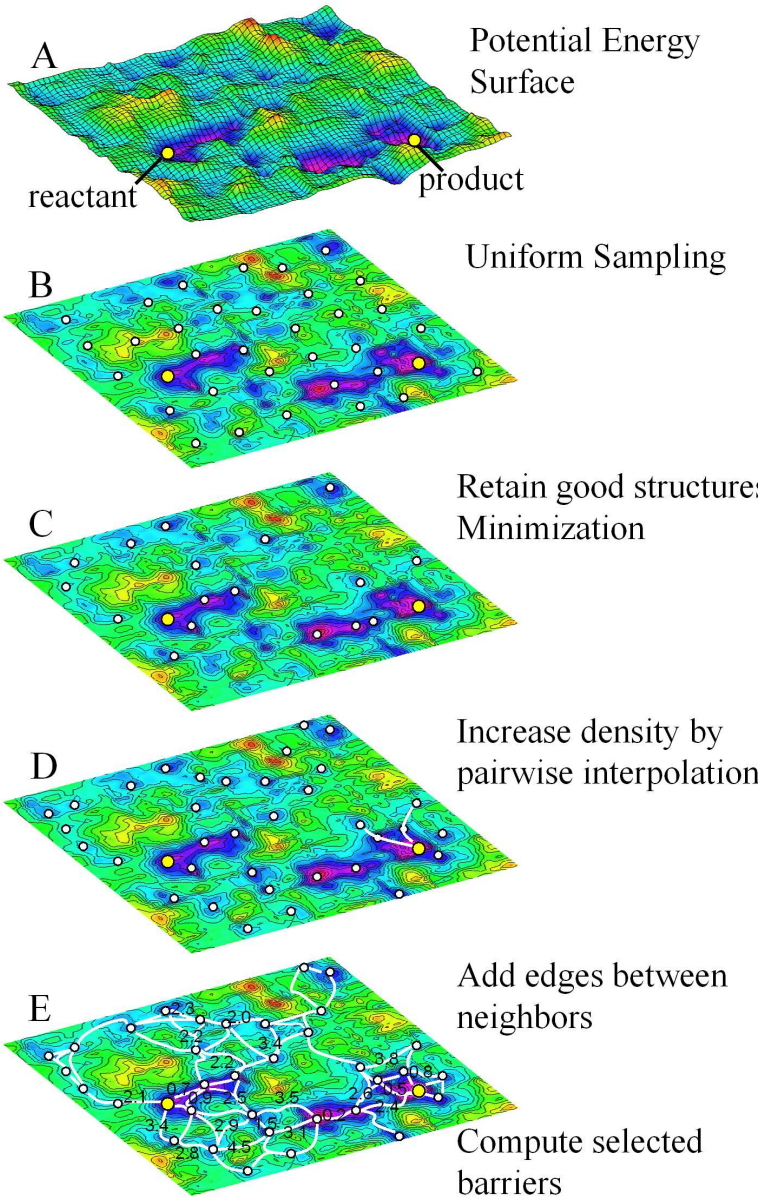


Figure 4.1: Overview of the steps involved in generating the Transition Network (TN). (A) The potential energy surface and the minimized reactant and product end-states of the transition. (B) Conformers are uniformly spread over the part of conformational space that is relevant to the transition (see Fig. 4.2 and Sec. 4.3), (C) Structures without steric clashes are accepted (see Sec. 4.4) and energy-minimized (see Sec. 4.5). (D) More low-energy minima are generated by pairwise interpolation between available conformers and minimization (see Sec. 4.6). These minima form the TN vertices. (E) Pairs of neighboring minima are associated, forming the TN edges. The sub-transitions barriers of selected edges are computed (see Chapter 3).

4.1 The Sampling(S) and Interpolation(I) Regions

Uniform sampling of all degrees of freedom in the protein is not desirable when studying conformational transitions in the native state. Most conformational transitions are relatively local in the sense that the major part of the protein retains its native fold. While there might be complex rearrangements in certain regions, even refolding of parts of the backbone (such as in Ras p21), the remainder of the protein only adapts flexibly to these changes. Therefore the protein is partitioned into an Interpolation (**I**) and a Sampling (**S**) region (see Fig. 4.2a for an illustration). The rotable torsion angles of the **S**-region (including backbone ϕ/ψ angles and single-bond side-chain angles) constitute the degrees of freedom in which the conformational sampling is performed. The **I**-atoms respond flexibly to changes in the **S**-region, and can be seen as controlled by a reaction coordinate, such as the interpolation distance between the transition end-states. (see Fig. 4.2B for an illustration). Notably, the motion of the protein is not constrained to degrees of freedom defined by **S**, as all other degrees of freedom are locally relaxed in the minimizations following the sampling step.

Fig. 4.2A shows the **S** and **I** regions used here for Ras p21. The sampling (**S**) region was chosen to encompass the Switch I (residues 30-35) and Switch II (residues 61-70) regions.

4.2 Interpolation of I-Atoms

To obtain a smooth variation of the positions of atoms in the **I** region near the boundary to the **S** region, the coordinates of the **I** region are generated by interpolating between the end-states of the transition. For this, a combined interpolation procedure is used: First, the backbone atoms are interpolated in Cartesian coordinates so as to preserve the backbone fold, and then the side-chain atoms are built onto the interpolated backbone, using internal coordinate values that are interpolated between the end-state values. This interpolation method was shown to produce less distorted structures than Cartesian or internal coordinate inter-

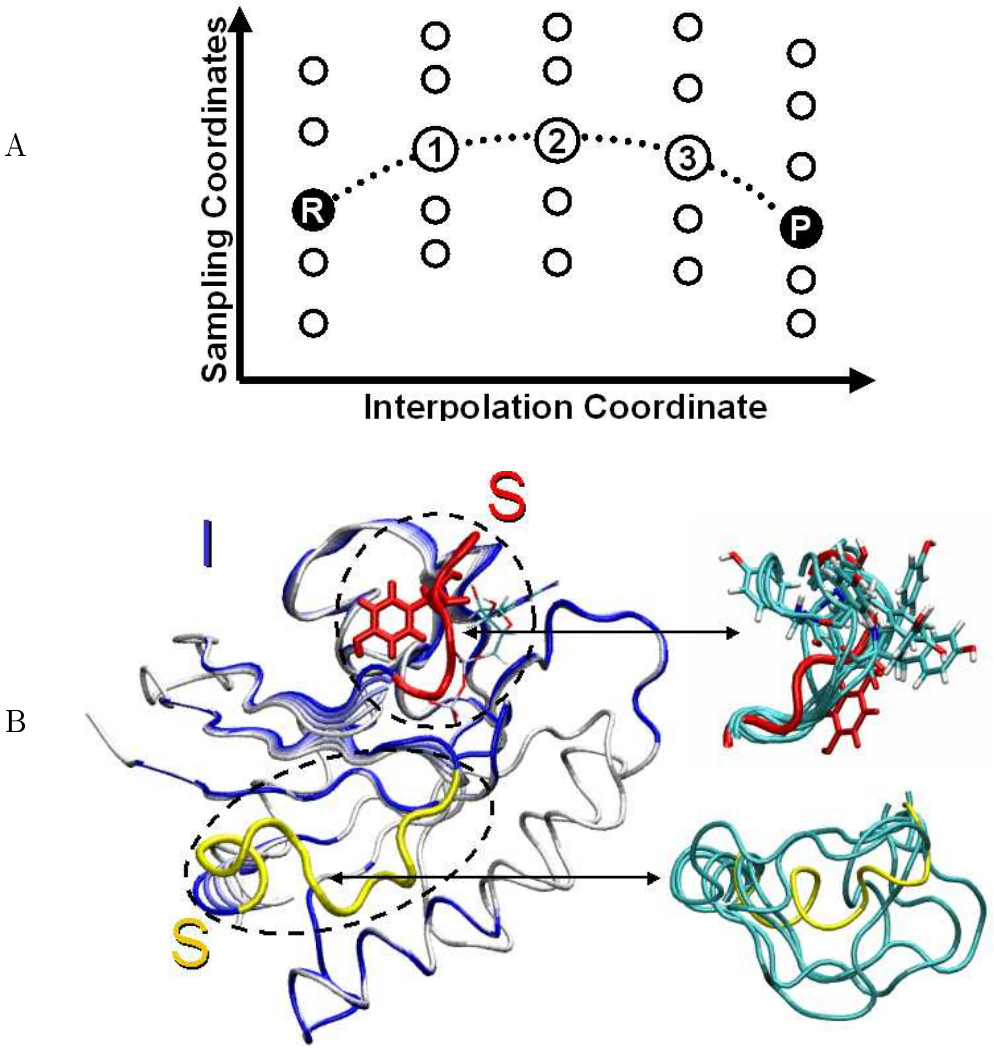


Figure 4.2: Illustration of the sampling procedure. (A) A number of intermediate conformations are generated by interpolating a subset of the protein atoms (the **I**-region) between their end-state positions ("R" and "P", here 3 intermediate structures are generated). From each interpolated structure, a large set of conformations is generated by sampling the torsional angles of the **S**-region of the proteins. (B) the interpolation (**I**) and sampling (**S**) regions. Left: The atoms of the **I**-region are interpolated between the transition end-states (shown in white and dark blue), here producing three intermediates (shown in shades of blue). Right: For the **S**-region (Switch I in red, Switch II in yellow), the single-bond torsion angles are sampled uniformly (examples of several **S** conformations are overlaid on the right side). The full set of conformations is defined by all combinations of the five conformations for the **I**-region with each sample of the **S**-regions.

polation alone [19].

For practical convenience, the combined interpolation is done for all atoms of the protein (including the atoms of the **S** region). Because the **S** region has by definition very different conformations in the end-states, the interpolated structures involve distorted internal coordinates in the **S** part of the backbone. To start the **S** sampling with reasonable values of the internal coordinates of the **S** region, each interpolated structure is energy minimized with positional harmonic constraints on the **I** atoms (force constant 1 kcal mol^{-1} Å^{-1}). In the example treated here, n_{interpol} =3 interpolated structures of Ras p21 were generated in this way, yielding 5 structures along the interpolation including the endstates.

4.3 Conformational Sampling of the S Region

For each of the interpolated structures and the two end-states, a uniformly distributed set of conformers of the **S** region is generated (see Fig. 4.1b). Sampling of the **S** region is performed uniformly in the space of flexible torsion angles, comprising the ϕ/ψ backbone and single-bond side-chain angles. The stiff internal degrees of freedom (*i.e.* bond lengths, valence angles and backbone ω angles) are not sampled here.

If the sampling region is located at one of the termini of the polypeptide chain, there are no closure constraints on the backbone, allowing the ϕ/ψ angles to be sampled directly by setting them to random values. When the sampling region is an intermediate part of the polypeptide chain (as is the case for the Switch I and Switch II loops in Ras P21), this "free" sampling is not possible, as it would violate backbone closure (*i.e.* backbone bond lengths and angles would not be preserved).

4.3.1 Generation of Backbone Conformers with a Window Method

Random backbone conformations were generated using a variant of the window method proposed in [75, 76]. This procedure allows backbone variation of a se-

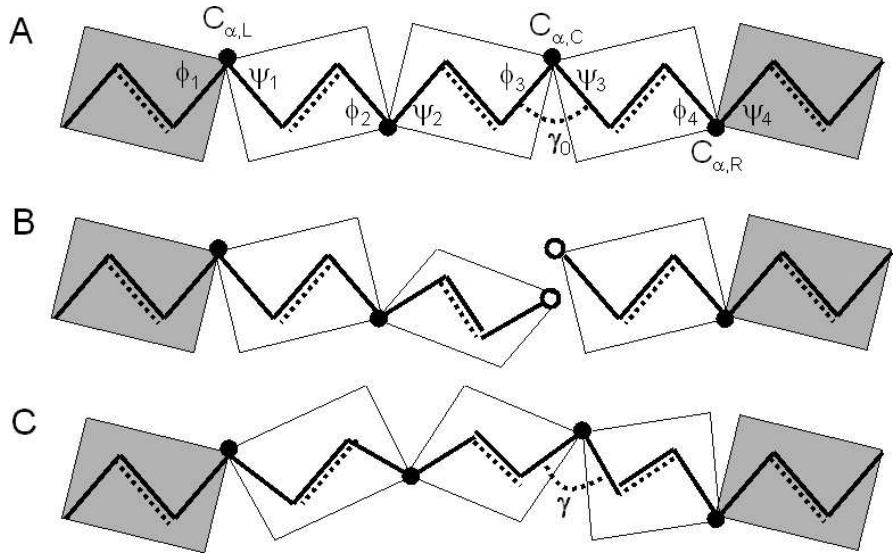


Figure 4.3: Illustration for a backbone window move. A) The conformation within the window (white) may vary while the window's boundaries (gray) have a fixed position. The flexible ϕ/ψ-torsions are labeled. The ω torsion angle around the peptide bond (dotted) gives rise to the stiff peptide planes (rectangles). B) The cut atom ($C_{\alpha,C}$) separates the left and the right arm, which are moved independently according to the pre-rotations defined by the user (here, at ϕ_2, ψ_2). C) The method rotates the ϕ/ψ-torsions at the joints such that backbone closure and the γ-valence angle is restored at the cut. Thereby, the method determines the six torsion angles at the joints and at the cut (here: ϕ_1, ψ_1, ϕ_3, ψ_3, ϕ_4, ψ_4).

ries of $r \geq 3$ consecutive residues (the *window*) while preserving the boundary condition of that window. Out of the window's $2r$ ϕ/ψ torsion angles, $2r-6$ can be freely chosen and pre-rotated. The remaining six torsion angle values are being determined by the method, as six degrees of freedom are required to define translational and rotational invariance of the window boundaries.

The C_α's of the first and last residue in the window are here termed *left joint* $C_{\alpha,L}$ and *right joint* $C_{\alpha,R}$, respectively. One of the C_α's between the joints is selected as *cut atom* $C_{\alpha,C}$. The set of atoms between $(C_{\alpha,L}, C_{\alpha,C})$ and $(C_{\alpha,C}, C_{\alpha,R})$ are termed *left* and *right arm*, respectively. See Fig. 4.3 for an illustration.

The pre-rotations are defined by assigning random values to all $2r-6$ ϕ/ψ-angles inside the window, excluding those at $C_{\alpha,L}, C_{\alpha,C}$ and $C_{\alpha,R}$ [1]. Here, the atoms of

[1]Proline ϕ angles are chosen in the range $[-80,-40]$ degrees, while all other ϕ/ψ-angles may vary in the range $[-180,180]$

the two arms are rotated independently, *i.e.* one introduces an imaginary cut at $C_{\alpha,C}$, splitting it into a 'left' and 'right' $C_{\alpha,C}$. Rotations around bonds in the left/right arm are only propagated up to the left/right $C_{\alpha,C}$.

The six remaining degrees of freedom are used to correct the backbone conformation inside the window such as to restore the backbone closure. That is, the arms need to be rotated about their joints in such a way as to connect the left and right $C_{\alpha,C}$ while restoring the valence angle $\gamma = \measuredangle(N_C, C_{\alpha,C}, C_C)$. If the sum of the arm lengths is smaller than the distance $|\mathbf{x}_{C_{\alpha,L}} - \mathbf{x}_{C_{\alpha,R}}|$, then there is no solution for this set of pre-rotations because the arms are too short to restore backbone closure. Otherwise, the new position of $C_{\alpha,C}$ may lie on a circle which is perpendicular to the axis through the joints, having its center at $\mathbf{x}_c$:

$$\mathbf{x}_c = \frac{|\mathbf{x}_{C_{\alpha,L}} - \mathbf{x}_{C_{\alpha,C}}|^2 - |\mathbf{x}_{C_{\alpha,C}} - \mathbf{x}_{C_{\alpha,R}}|^2}{2|\mathbf{x}_{C_{\alpha,L}} - \mathbf{x}_{C_{\alpha,R}}|^2} \tag{4.1}$$

and a radius R of:

$$R = \sqrt{|\mathbf{x}_{C_{\alpha,C}} - \mathbf{x}_{C_{\alpha,L}}|^2 - |\mathbf{x}_c - \mathbf{x}_{C_{\alpha,L}}|^2} \tag{4.2}$$

In general, not all points on the circle are permitted for the new $C_{\alpha,C}$ positions, because only certain circle sections (up to 4) contain points of possible backbone closure (see Fig. 4.4).

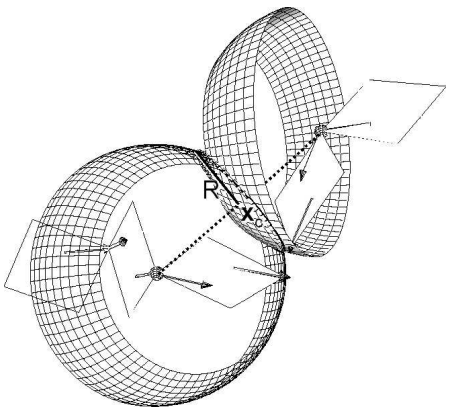


Figure 4.4: By rotating the two rigid pre-rotated arms around their joints, the left and right $C_{\alpha,C}$ reach points on an *iris*, *i.e.* the section of a sphere (a full sphere would be accessible only if the ϕ and ψ angles at each of the cuts would have a 90°-angle relative to each other). Both irises intersect at the cut circle, on which the new $C_{\alpha,C}$ is located. Figure taken from [75].

To determine the allowed regions for $C_{\alpha,C}$, a line-search is performed on the cut-circle. One point every 10° is tested for a solution by positioning $C_{\alpha,C}$ there. Now the positions of the second backbone atom in the window, C_L, and the second-last one, N_R, are computed. To do this, one takes into account that the distances

$|\mathbf{x}_{N_L} - \mathbf{x}_{C_L}|$, $|\mathbf{x}_{C_{\alpha,L}} - \mathbf{x}_{C_L}|$, and $|\mathbf{x}_{C_L} - \mathbf{x}_{C_{\alpha,C}}|$ remain unchanged during the whole process, and uses triangulation to compute $\mathbf{x}_{C_L}$ from $\mathbf{x}_{N_L}$, $\mathbf{x}_{C_{\alpha,L}}$, and $\mathbf{x}_{C_{\alpha,C}}$ [76]. $\mathbf{x}_{N_R}$ in the right arm is computed in an analogous way. If both triangulations give solutions, the tested point $\mathbf{x}_{C_{\alpha,C}}$ is a position that recovers backbone closure. For each such point, there are up to four possible backbone conformations, because each triangulation delivers two (typically different) solutions.

As a further requirement, a solution must fulfill:

$$f := \gamma - \gamma_0 = 0. \tag{4.3}$$

Because of the four different solutions for $\mathbf{x}_{C_{\alpha,C}}$, f has four branches on all sections of the cut-circle. To identify possible solutions of f, one identifies, for each branch of f, pairs of adjacent points $(i, i+1)$ on the cut-circle which give a solution for $\mathbf{x}_{C_{\alpha,C}}$ and which have the property sign(γ_i) = -sign(γ_{i+1}). If all intermediate points give a solution for $\mathbf{x}_{C_{\alpha,C}}$, then there is also at least one such solution with $f = 0$. This solution is approached by subsequently subdividing the interval between $i, i+1$.

Given the up-to-4 sections on the cut-circle which yield a solution for $\mathbf{x}_{C_{\alpha,C}}$ and the four branches of f, a total of up to $4^2 = 16$ correct solutions can be found for $\mathbf{x}_{C_{\alpha,C}}$. For each solution the remaining backbone atoms are then rebuilt (their positions are uniquely determined) and the side-chains are rigidly translated and rotated to their new position. The method returns all conformations that were found for a given set of pre-rotations.

4.3.2 Uniform Generation of Backbone Conformers

The above sampling method is repeated, each time choosing a random location and length of the window in the **S** region and its pre-rotation. A solution is considered "valid" if the resulting backbone atoms do not collide with the **I** region of the protein and if atoms whose positions are directly determined by the backbone configuration (*i.e.* backbone O and H, H_α, C_β, Proline side-chains) can be placed without collision. Sec. 4.4 describes how these collision checks are performed.

To obtain a conformational sample that is approximately uniform, conformers that

are valid (*i.e.* have no collisions) are "accepted" (*i.e.* added to a "conformational repository") only if they significantly differ from already-accepted conformers as measured by the ϕ/ψ dihedral RMS difference, which must exceed a chosen value, δ_s. The choice of δ_s determines the density of sampling. It is set according to how finely the details of the transition should be probed. Backbone conformers are generated until the sampling density defined by δ_s has been reached. The criterion used here for this is that no "valid" structure are "accepted" anymore for a number of n_{reject} successive attempts. This yields a number of n_i^{back} backbone conformations for each interpolation step i ($i \in \{0, 1, ..., n_{\text{interpol}}+1\}$, where $i = 0$ and $i = n_{\text{interpol}} + 1$ are associated with the end-states and 1, ..., n_{interpol} are the interpolated intermediates).

For Ras p21, the Switch I and II **S**-regions were sampled independently, using $\delta_s = 50°$ and $n_{\text{reject}} = 1000$. For each interpolation step, i ($i \in \{0, ..., 4\}$), this yielded $n_i^{\text{back1}} \approx 30$ backbone conformers for Switch I and $n_i^{\text{back2}} \approx 10^4$ backbone conformers for Switch II. This difference can be explained by the different number of free backbone dihedrals in Switch I ($2 \cdot 6 - 6 = 6$ free ϕ/ψ angles) and Switch II ($2 \cdot 10 - 6 = 14$ ϕ/ψ angles). A backbone conformer was generated by randomly combining a Switch I with a Switch II conformation associated with the same interpolation intermediate. This yields a total number of 1.5×10^6 possible backbone conformers.

4.3.3 Uniform Generation of Side-chain Conformers

To obtain a complete conformation, the side-chains of the **S** region are build onto a randomly picked backbone out of the n_i^{back} generated backbones, using randomly-chosen single-bond torsion angles. The resulting conformer is accepted if it does not involve atom collisions (see Section 4.4), giving for each interpolation step i a number n_i^{full} of sterically valid conformations of the **S** region. $n_i^{\text{full}} = k^{\text{side}} n_i^{\text{back}}$, where k^{side} is the desired average number of sidechain conformers per backbone conformer. This trivial method is not very efficient in practice, firstly because some backbone conformers may never allow a given side-chain to be built without collisions, and secondly because for a given backbone conformer it is unlikely that placing all sidechains at once produces a conformation without collisions. Here, a more efficient method is used that consists of the following steps: 1) For each

backbone conformation c, a weight w_c is computed which is equal to the probability that a set of noncolliding sidechains can be built on this backbone, when a uniform distribution of sidechain torsion angles is used. 2) A random backbone conformation is selected according to the probability $p_c = w_c / \sum_k w_k$. 3) Onto the selected backbone, each sidechain is build by itself in a number of conformations that do not produce collisions with the backbone and the non-sampled regions of the protein. 4) Side-chain conformations from step 3 are combined randomly to form a fully build protein conformation, which is accepted if is does not have any collisions. Steps 2-4 are repeated until a desired number of conformations have been generated.

The weight w_c is computed as follows: For each backbone conformation c, an acceptance probability $p_{c,i}$ for each side-chain i is calculated by generating a large number of random rotamers for that side-chain (in the absence of the other side-chains of the **S** region) and counting the number of non-colliding rotamers. If any $p_{c,i} = 0$ (*i.e.* some side-chain cannot be placed at all without producing collisions), then backbone conformation c is permanently rejected and $w_c = 0$. Otherwise, the probability q_c to find a noncolliding combination of the individually valid sidechain conformations is computed. This is done by generating a large number N_c of random combinations of valid side-chain rotamers and counting the number n_c of non-colliding combinations, $q_c = n_c/N_c$. The weight w_c is obtained as $w_c = q_c \prod_i p_{c,i}$.

For Ras p21, an average of $k_{\text{side}} = 10$ side-chain conformations per backbone conformer were generated, yielding $n_i^{\text{full1}} \approx 300$ and $n_i^{\text{full2}} \approx 10^5$ collision-free conformations of Switch I and II, respectively. Combining pairs of these Switch I and II conformers yielded 3×10^7 fully build protein structures for each interpolation step i. Thus, the total number of collision-free and significantly different structures is $n^{\text{full}} = 1.5 \times 10^8$, forming a large conformational repository from which structures can be drawn and further energy optimized. The conformations in this repository are distributed uniformly within the sterically accessible regions of the conformational subspace spanned by the torsional coordinates of **S** and the interpolation coordinate of **I**.

4.4 Validating Conformers in the Initial Sample

During the sampling procedure described above, new conformers are validated by checking that they do not produce very high potential energies (see Fig. 4.1C). For this, conformations are rejected if they involve atom collisions. A collision between a pair of atoms i, j is defined such that the Lennard-Jones and Coulomb interaction energy of that pair exceeds a tolerance value E_{tol} ($E_{tol} = 20\text{kcal}\,\text{mol}^{-1}$ was used in this study). The collision check needs to be repeated so often that it is a computational bottleneck for the sampling method. To avoid computing the interaction energies for all pairs, we compute a minimum allowed distance $d_{i,j}^{\min}$ for each atom-pair, which is the root of following equation:

$$\epsilon_w \left[\left(\frac{\sigma}{d_{i,j}^{\min}} \right)^{12} - \left(\frac{\sigma}{d_{i,j}^{\min}} \right)^{6} \right] + \frac{q_i q_j}{4\pi\epsilon_0 d_{i,j}^{\min}} - E_{tol} = 0, \tag{4.4}$$

where ϵ_w is the van der Waals well depth, σ_{ij} is the effective van der Waals radius for atoms i and j, q_i and q_j are the partial charges of atoms i and j and $\epsilon_r\epsilon_0$ is the dielectric constant. Above equation is solved for $d_{i,j}^{\min}$ with Newton's root-finding method. For the E_{tol} used in this study, there was always a unique solution for $d_{i,j}^{\min}$. If smaller E_{tol} are used, Eq. (4.4) may have two solutions, in which case, the smaller solution must be used so as to assure that $d_{i,j}^{\min}$ reflects the repulsive interaction. The resulting $d_{i,j}^{min}$ values are stored. A given conformation is treated as valid if all non-bonded atom distances, $d_{i,j}$ (excluding 1-4 pairs) fulfill the criterion $d_{i,j}^{min} \leq d_{i,j}$. The number of distance computations is kept small by embedding the protein coordinates in a lattice and computing distances only between atoms which have been changed in a given sampling step and atoms which are in the same or adjacent lattice cells.

4.5 Minimizing the Conformers

To obtain a representative collection of low-energy minima, a number of $n_{\min}$ conformers is drawn randomly from the conformational repository and energy-

minimized on the potential $U(\mathbf{x})$ (see Fig. 4.1c). Only minima which reach a low-energy region defined by $U(\mathbf{x}) < E_{\text{low}}$ are accepted, where E_{low} is a predefined constant. Minimization of many conformers is expensive, so it is desirable to reject structures early which are not likely to fall into low-energy minima. An efficient method to do this is to reject high-energy minima can based on statistics set up during a number of full minimizations. The following method is used here: For Ras p21, 100 samples were retrieved from the sample repository and fully minimized to a gradient RMS of 10^{-5} kcal mol^{-1} Å^{-1}. Each of these minimization trajectories delivered a series of gradients $(\mathbf{g_0}, \mathbf{g}_1, ..., \mathbf{g}_n)$ and associated potential energies $(U_0, U_1, ..., U_n)$, where the pair $(\mathbf{g}_n, U_n)$ corresponds to the fully minimized structure. All tuples $(\mathbf{g}_i, U_i)$ from all 100 minimization trajectories were used to set up a correlation statistics between $\mathbf{g}_i$ and $\Delta U = U_i - U_n$, *i.e.* the energy difference from the fully minimized structure. These statistics, shown in Fig. 4.5, were used to obtain for each range of gradient values a corresponding value of ΔU that was higher than 90% of the ΔU's in that range. This yields an upper estimate of ΔU, given a certain gradient $\mathbf{g}$. This estimate was used to reject structures during minimizations if their minimum energy, predicted from this upper estimate, considerably exceeded the energy tolerance threshold: $U(\mathbf{x}) - \Delta U > E_{\text{low}} + 10$ kcal mol^{-1}.

For Ras p21, 15000 conformers were randomly retrieved from the sample repository out of which 189 reached the low-energy region (E_{low} was defined to be 50 kcal/mol above the reactant energy). These were minimized to a gradient RMS of 10^{-3} kcal mol^{-1} Å^{-1} and formed a low-density set of conformations in the low energy region $E(\mathbf{x}) < E_{\text{low}}$.

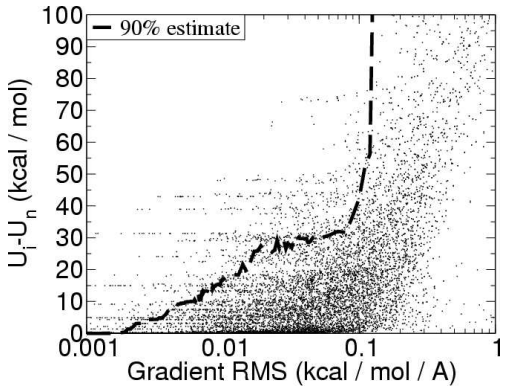


Figure 4.5: Using the gradient during minimizations to predict the expected energy at the minimum. Based on the minimizations of 100 different conformers, each minimization going through a series of intermediates with gradients $(\mathbf{g}_0, ..., \mathbf{g}_n)$ and energies $(U_0, ..., U_n)$, the difference between the energy of an intermediate and the final (minimum) energy, $U_i - U_n$ is plotted against the current gradient $\mathbf{g}_i$. 90% of the points are below the dashed line, which can be used to estimate how much more the energy may decrease during a minimization, based on the current gradient value, thus allowing to abort non-promising minimizations.

4.6 Increasing the Low-Energy-Conformer Density

Increasing the density of conformers in the low-energy regions can, in principle, be done by minimizing more structures from the conformational repository. Given the low yield of this approach (see above: 189/15000$\approx$ 1.25%), this is computationally inefficient. Instead, additional conformers are built by interpolation between the already-found low-energy conformers. This can be done in various ways. The strategy used here was to select each pair of low-energy conformers separated by a distance in the range $\delta_{\min}^{\text{interpol}} = 0.75$ Å and $\delta_{\max}^{\text{interpol}} = 2$ Å (measured as Cartesian RMSD of the C_α atoms in the **S** region) and to generate an interpolation pathway between them using the method described in Sec. 4.2. Two structures were generated, one third and two thirds of the way along each interpolation, respectively, and energy minimized as described in Sec. 4.5 (see Fig. 4.1d). This procedure was efficient in finding low-energy minima, increasing the number of conformers below E_{low} from 189 to 10831. This considerably increased the average number of neighbors for each minimum from 3 to 267 ("neighborhood"

being defined by a cutoff distance $\delta_{\max}^{\text{connect}}$).

During minimization, it is possible that some conformers end up in similar minima. This produces conformational redundancy, which was subsequently removed. For this, minima were considered in the order of increasing energy, accepting only those minima whose nearest-neighbor distance to any already-accepted minimum was at least $\delta_{\min}^{\text{connect}} = 0.75$ Å . This led to a final number of $|\mathcal{V}| = 6242$ diverse minima which served as the vertices of the transition network.

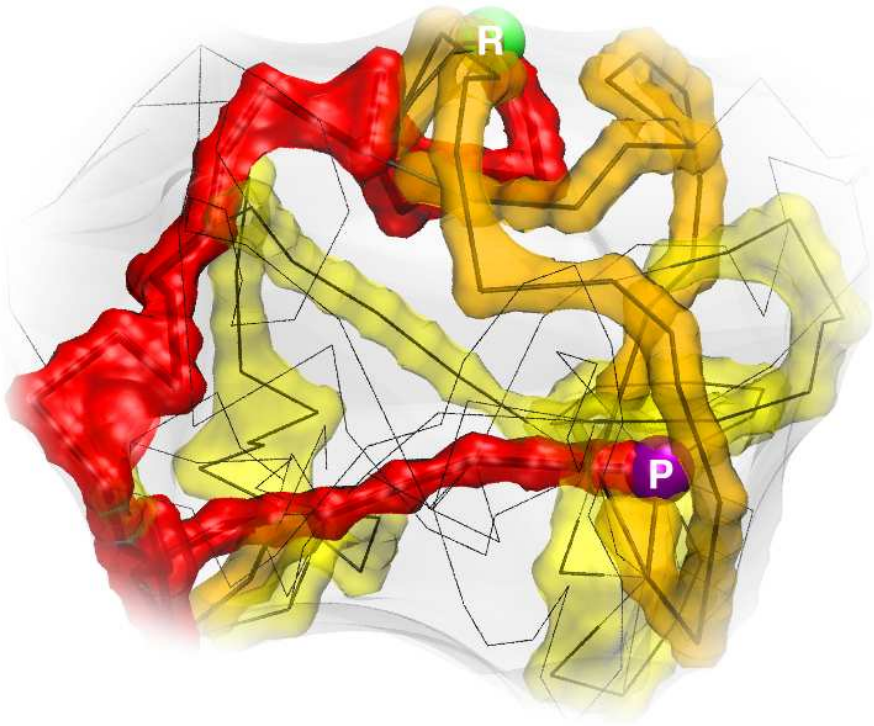
4.7 Conclusion

The sampling method presented in this chapter allows to uniformly sample conformations for complex conformational changes in proteins. Although it does not generate a statistical ensemble which obeys the Boltzmann distribution, it allows to efficiently explore the conformational space relevant to the conformational change without being hindered by energy barriers. Statistical ensembles, which allow for the computation of free energies, probabilities and rates can be computed later for certain regions of interest, defined by a subset of the conformational sample (see also Sec. 2.1.3 and the discussion in Chapter 6).

The conformational sample obtained here allow to generate a static TN for Ras p21 based on potential energies. Together with the algorithms introduced in Chapter 3, it enables a comprehensive analysis of the Ras p21 conformational switch, which is given in the next chapter.

Chapter 5

Comprehensive Analysis of the Ras p21 Conformational Switch



Ras p21 is a protein that can switch between two stable conformations in order to communicate a growth signal (or the lack thereof) to the interior of the cell [6, 77]. Although the end-states of the switch are known from X-ray crystallography [78, 79], no experimental evidence of the actual mechanics of the switch transition is available. The switch is extremely slow (the rate was measured to be on the order of 10^{-4} s^{-1} [80], *i.e.* one event per every few hours), rendering standard molecular dynamics simulation useless. The switch transition is also very complex, involving a rearrangement of parts of the backbone that include more than 30 flexible torsion angles, so that no unambiguous guess of the reaction pathway is possible. Nevertheless, previous studies have attempted to describe this transition using methods which rely on such an initial guess [19, 81, 82], yielding the main result that the switching process is by far too complex to be captured by a single reaction pathway. In the present chapter, we demonstrate the capability of the methods developed in this work to model complex molecular processes by giving a comprehensive Transition Network analysis of the Ras p21 conformational switch that yields valuable insight into its mechanism. Among the questions that have been raised by a previous study of this conformational transition [19] and that are addressed here are: 1) Is the rearrangement of Switch I characterized by the side-chain of Tyr32 threading underneath the backbone or by moving it through the solvent (see Fig. 5.1)? 2) Is there a coupling between the Switch I and Switch II transitions, *i.e.* is the relative order of events in the two Switch regions strictly defined? 3) Is there a well-defined unfolding pathway of Switch II?

The computational methods presented in Chapter 3 were used to compute best transition pathways (Sec. 5.2) and energy ridges (Sec. 5.3) for the conformational switch, demonstrating the order of complexity that can be accessed with these methods (The system setup is described in Appendix C). The analysis of the best paths and the energy ridges, together with a residue-based energy decomposition of the transition states (Sec. 5.4), yields extensive information on the mechanism of the conformational switch and give answers to the questions posed above.

In lack of reliable methods to compute appropriate free energies for TN vertices and edges (see Sec. 2.1.3), the Ras p21 TN is based on potential energies only. The results shown in this chapter should therefore be treated in the sense of Eq. 2.10: Comparing alternative pathways or transition states is based on a correlation. A large energy difference $\Delta E_{\text{pot}} \gg k_B T$ ($k_B T \approx 0.6$ kcal/mol at $T = 300$ K)

between two alternative pathways gives clear preference to the lower-energetic pathway, while energetically similar pathways cannot be reliably distinguished and the energy barriers cannot be directly transfered into rates (see also Sec. 2.2.2).

5.1 The Functional Cycle of Ras p21

Before going into the details of the molecular switch it is useful to have an overview of the functional context. The intrinsic[1] functional cycle of Ras p21 consists of four main states, in which Ras is switched between an active (Ras*) and inactive (Ras) form:

$$\begin{array}{ccc} \text{A)}\, \mathrm{Ras^* \cdot GTP + H_2O} & \rightarrow & \text{B)}\, \mathrm{Ras^* \cdot GDP \cdot P_i} \\ \uparrow & & \downarrow \\ \text{D)}\, \mathrm{Ras + GDP + P_i} & \leftarrow & \text{C)}\, \mathrm{Ras \cdot GDP + P_i} \end{array}$$

A) In the active form Ras p21 is bound to the molecule guanosine-tri-phosphate (GTP) and enables cell growth.

B) The hydrolysis reaction $\mathrm{GTP + H_2O \rightarrow GDP + P_i}$ has "discharged" GTP into guanosine-di-phosphate (GDP) and inorganic phosphate ($\mathrm{P_i}$).

C) After the conformational change and the release of $\mathrm{P_i}$, Ras is in the inactive form $\mathrm{Ras \cdot GDP}$.

D) When GDP has been released to the solvent, Ras is in an (structurally unknown) inactive form. It is ready to be recharged by GTP, and to re-enter the cycle in step A.

Fig. 5.1 illustrates the functional cycle of Ras. Cycle step B→C involves the conformational switch that is modeled here. A critical process is the release of the $\mathrm{P_i}$, which might, in principle, occur before, after, or during the conformational change. To test which of the three options are the most likely, the energetics of the $\mathrm{Ras \cdot GDP \cdot P_i}$ and $\mathrm{Ras \cdot GDP + P_i}$ networks were compared. A sampling of the

[1] "intrinsic" means that the cycle only involves Ras and its ligands. *In vivo*, steps A → B and C → D → A are normally accelerated by the interaction between Ras with two other proteins: the GTPase Activating Protein (GAP) catalyzes GTP hydrolysis and the Guanine Nucleotide Exchange Factor (GEF) catalyzes the nucleotide exchange.

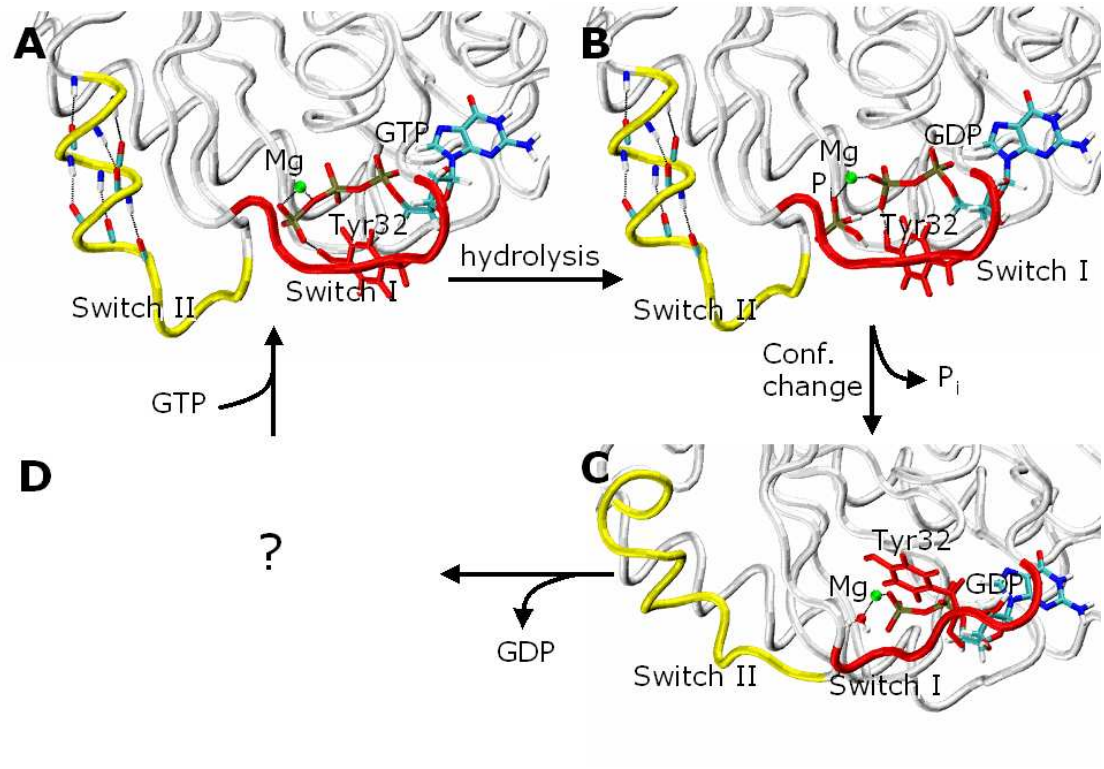


Figure 5.1: Structures in the intrinsic functional cycle of Ras p21. A) Ras p21 is active when it is bound to GTP (pdb entry 5p21, [78]). B) Immediately after hydrolysis ($GTP + H_2O \rightarrow GDP + P_i$) Ras is still in an (unstable) active form, but GDP and inorganic phosphate, P_i, are bound to it. C) The release of the inorganic phosphate from the binding site is associated with a conformational change. In this *conformational switch* of Ras p21, the Switch I loop rearranges such that Tyr32 moves to the other side of the backbone, and the Switch II helix unfolds. After the switch has completed, Ras p21 enters its inactive GDP-bound form (pdb entry 1q21, [79]). D) When GDP leaves the binding site, Ras is in an inactive form in which no nucleotide is bound (no experimental structure available). To become active again, Ras must be "recharged" with GTP.

conformational subspace relevant for the switch was conducted for both systems. The resulting TN for $\text{Ras} \cdot \text{GDP} \cdot P_i$ had $|\mathcal{V}|$=8445 and the one for $\text{Ras} \cdot \text{GDP}$ (sampling described in Chapter 4) had $|\mathcal{V}|$=6242 vertices. The connectivity of both TN was defined by connecting all vertices-pairs by an edge if their distance (measured as RMS distance of the C_α-atoms of the **S** regions) was less than 1.5 Å. To have an *a priori* definition of the edge energies, the mean value from the energy barrier statistics given in Appendix C.3 was used.

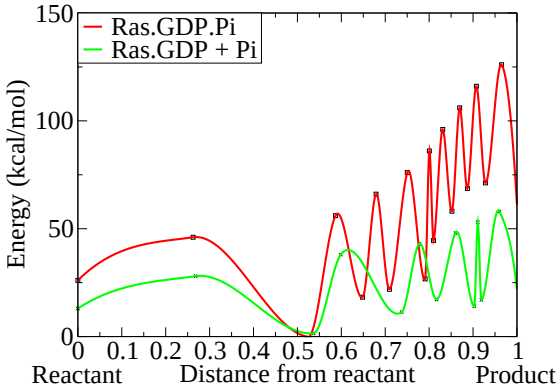


Figure 5.2: Energy profile for the Ras · GDP · P_i and Ras · GDP + P_i systems. The profiles were generated by "flooding" the TN starting from the reactant structure (see text). For each flooding energy, this procedure yields a set of vertices that are newly explored, denoted by the energy minima in the plot. The barriers between pairs of these vertex-sets are given by the energy maxima in the plot.

An "energetic flooding" was performed for both TN starting from the reactant vertex (the active form, see also Sec. 4.6). For this, we increased the flooding energy, starting from the reactant energy, gradually in steps of 5 kcal/mol. In each step, all vertices $\mathcal{V}_{new}$ that are newly reached by overcoming barriers not higher than the flooding energy, are marked. The set $\mathcal{V}_{new}$ is assigned an energy equal to the minimum energy of its vertices, and a distance equal to the mean distance between the reactant and each of its vertices (where the distance is measured as RMS difference between the C_α-atoms of the **S** regions, normalized to the distance between reactant and product). Barriers between the subsequent vertex-sets are defined by taking for their energy the flooding energy that needs to be overcome to reach a set and a distance is assigned to each barrier by the average of the distances of the previous and the next vertex sets. In this way, an energy profile as plotted in Fig. 5.2 is obtained[2].

Comparing the energy profiles shows that the Ras · GDP + P_i profile is flatter and lower than the Ras · GDP · P_i profile. The product energy on the Ras · GDP · P_i profile exceeds the rate-limiting barrier for the full transition on the Ras·GDP+P_i profile, such that it is unlikely that the inorganic phosphate, P_i, is released after

[2]In either TN setup, P_i or H_2O was in the γ-phosphate place of the magnesium ion coordination sphere. Thus, the systems for which energies were computed are a) Ras · GDP · P_i and b) Ras · GDP · H_2O. To allow for a meaningful comparison, they are transformed to a') Ras · GDP · P_i + H_2O and b') Ras · GDP · H_2O + P_i by adding the solvation energies and translational entropies of H_2O to the Ras · GDP · P_i TN ($E^{H_2O}_{solv} - TS^{H_2O}_{trans} \approx -20.5$ kcal/mol) and P_i ($E^{P_i}_{solv} - TS^{P_i}_{trans} \approx -142$ kcal/mol), respectively. The solvation energy is the free energy difference gained by putting the compound from vacuum into water and was computed with the ACE 2 implicit solvent model[83]. The translational entropy was computed at T=300K and a pressure of 1 bar.

the conformational change. The second energy barrier on the Ras·GDP·P_i profile also exceeds this rate-limiting barrier, suggesting that P_i is released before, or in an early step, of the conformational switch and the conformational change occurs (at least mainly) on the Ras · GDP + P_i surface.

Consequently, all further calculations were performed using the Ras·GDP TN (P_i is assumed to have left to the solvent already and is not treated as a part of the system). To reduce the complexity of the network, the number of neighbors for each vertex was restricted to the 20 nearest neighbors. The resulting transition network had $|\mathcal{V}|$=6242 vertices and $|\mathcal{E}|$=47404 edges.

The "reactant" and "product" vertices were redefined on the network by selecting the lowest energy minima within the vicinity of the crystallographic reactant and product structures after quenched MD (see Appendix C). The "vicinity" was defined here to be within both a ϕ/ψ-RMSD of 50° and a Cartesian RMSD of 1.5 Å for the C_α-atoms of the Switch regions. The resulting Cartesian RMSD over all C_α atoms between the crystal structures and the so-chosen reactant and product conformers was 1.4 and 1.5 Å , respectively.

5.2 Best Paths

Best paths between the reactant and product structures of the Ras p21 conformational switch (defined in Sec. 4.6) were computed using the iterative algorithm described in Sec. 3.2.1, and then structurally analyzed.

5.2.1 Computational Effort

To evaluate the performance of these algorithms on the Ras p21 TN, *a priori* values for the upper and lower bounds on edge barriers (*i.e.* $B_{uv}^{\min} = 0$ and $B_{uv}^{\max} = \infty$), as well as statistically estimated bounds (described in Appendix C.3) were used. We also examined the partial computation of best paths, in which only the best-path-sections within an energy interval ΔE_{sure} below the rate-limiting steps are determined (see Sec. 3.2 and Fig. 3.9A). Table 5.1 shows the number of edges required to be computed with CPR, n_{ec}, for the best path under these different conditions to be determined (starting the count from scratch

for each setting).

To determine the whole best path using *a priori* bounds on the energy barriers required nearly four times as many computations (n_{ec}=2252) as to compute it using statistical estimates of the bounds (n_{ec}=603). This faster convergence behavior agrees to our results for random TN in Sec. 3.4.3, demonstrating that the computation time for best paths can be greatly reduced by introducing a small uncertainty. The maximum error on the rate-limiting barrier resulting from the present estimates of $E_{uv}^{\min}$ would be 5.25 kcal/mol (from Eq. 3.6), but here, statistical estimation determined the correct best path nearly completely. Indeed, except for one additional, insignificant low-energy edge, the estimated best path is equal to the true best path. In particular, the rate-limiting energy barriers are the same.

The computational savings are even larger when the determination of the best path is limited to its highest-energy barriers. The value of n_{ec} when only the highest barrier along the path is certain ($\Delta E_{\text{sure}} = 0$) are compared with n_{ec} when all barriers along the best path are determined ($\Delta E_{\text{sure}} = \infty$). In conjunction with *a priori* bounds ($B_{uv}^{\min} = 0$, $B_{uv}^{\max} = \infty$), using $\Delta E_{\text{sure}} = 0$ reduces n_{ec} only slightly (from 2252 to 2059). However, in conjunction with statistical estimates on the edge energies, using $\Delta E_{\text{sure}} = 0$ reduces the n_{ec} by a factor of six (from 603 to 106). This is due to the fact that, when statistical estimates are used, many edges have an upper energy bound which is already below the energy region of interest ($E_{uv}^{\max} < E_{\text{peak}} - \Delta E_{\text{sure}}$), which means that their barriers do not need to be computed.

5.2.2 Structural Analysis

The 13 pathways of lowest energy (with rate-limiting barriers within a range of 10 kcal/mol) were computed and structurally analyzed. This showed that in the best path, about half of the Switch II helix first unfolds before the rearrangement of the Switch I, in which Tyr32 passes underneath the backbone. Subsequently, the rearrangement of Switch II completes. This latter step has the highest potential energy barrier along the best path (E_{peak} =45 kcal/mol relative to the reactant), indicating that it is rate-limiting. In these aspects, the 12 next best pathways

ΔE_{sure}[a]	Prior[c]	Est.[d]
0	2059	106
5	2059	114
10	2069	212
15	2115	321
20	2208	505
25	2224	565
30	2246	589
∞[b]	2252	603
Len.[e]	23	24
Barr.[f]	45.7	45.7

Table 5.1: Number of edges computed to determine the best path, assuming no barrier has been previously computed. Computing an edge on a 3 MHz PC took around 2 hours CPU time, on average. a) Energy range below the highest barrier within which lower barriers are known to belong to the best path (see Fig. 3.9A). b) The correct best path is determined. c) Using $B_{uv}^{\min} = 0$, $B_{uv}^{\max} = \infty$ as bounds on the unknown edge energy barriers. d) Using stat. estimates (Appendix C.3) to guess $B_{uv}^{\min}$ and $B_{uv}^{\max}$. e) Number of edges along the fully-determined best path. f) Rate-limiting energy barrier along path, in kcal/mol relative to reactant.

are similar, the differences between them mainly lie in the precise order of events in the Switch II unfolding (e.g. in the degree of Switch II unfolding at the time when Tyr32 passes underneath the backbone). A sketch of the best pathways is shown in Fig. 5.3. The timescale of the Ras p21 conformational switch[80] requires that no free energy barrier along the path exceeds 23 kcal/mol. This implies that a significant entropic contribution, due possibly to an increase of backbone flexibility, reduces the high enthalpic barrier found here.

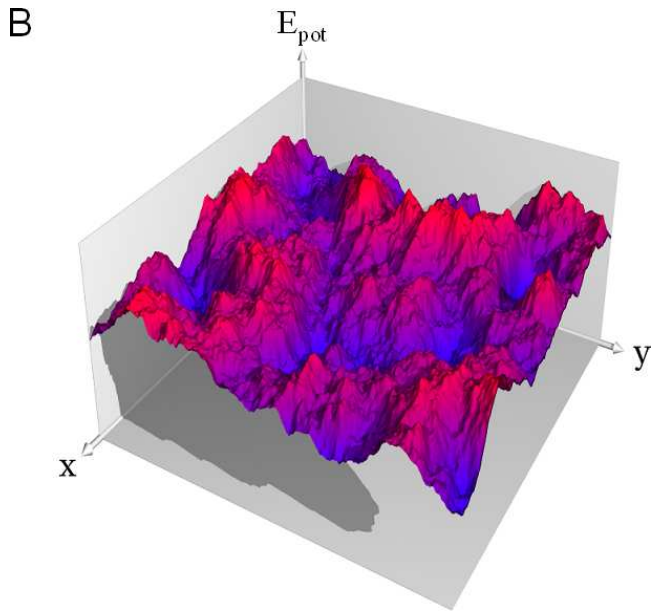


Figure 5.3: 2D representation of the potential energy surface of Ras p21. The horizontal and vertical axis measures, respectively, the orientation of Tyr32 on the Switch I loop (dihedral angle P_β,C_{32},N_{32},OH_{32}, in degrees), and the helicity of Switch II (number of α-helical H-bonds). The energies are those of the TN vertices and are encoded by color (dark gray=0 kcal/mol, light gray=60 kcal/mol). Reactant and product structures are shown by the 'R' and 'P' bullets. The triangles mark the rate-limiting transition state of the Switch I transition along low-energy paths and correspond to the lowest-energy points shown in Figs 5.8b and c. The best transition pathway is shown in white, the next-best transition pathways whose rate-limiting step is up to 10 kcal/mol higher are shown in yellow, red, magenta and cyan. The best path with Tyr32 moving through the solvent is shown in blue.

Is the rearrangement of Switch I necessarily associated with a passage of Tyr32 underneath the backbone? The energy barrier associated with the Tyr32 passage is 25 kcal/mol along the best path. For comparison, along paths in which Tyr32 passes the other way (*i.e.* through the solvent), the barrier of that passage is at least 40 kcal/mol. Although still lower than the rate-limiting barrier of the whole pathway, this clearly indicates that passage underneath the backbone is the preferred mechanism.

Is there a typical order of events in Switch II? To examine this, the α-helical hydrogen bonds between the Switch II residues 64 to 73 were evaluated along the 5 best paths. Fig. 5.4A shows which hydrogen bonds are present at different positions along these pathways. An α-helical hydrogen bond was defined as 'present',

if the distance between the backbone-O atom of residue r and the backbone-H atom of residue $r+4$ was ≤ 2.2Å. All 5 pathways exhibit a relatively similar behavior as to the order of rupture or formation of these hydrogen bonds: H-bond 69-73 (the hydrogen bond between residues 69 and 73) stays active along all best paths and is only temporarily broken or weakened to allow for rearrangements of neighboring residues. H-bond 68-72 is very weak in the reactant and becomes transiently active during the transition, stabilizing intermediate structures. H-bonds 65-69, 66-70 and 67-71 are lost at about half of the transition whereas H-bond 64-68 is lost very early in all paths and only transiently reformed for paths 3 to 5. The overall tendency is that the Switch II helix unfolding progresses from its N-terminal (nucleotide-near) to its C-terminal (nucleotide-far) end. This is also apparent in Fig. 5.4B which shows four structures of Switch II along the best path.

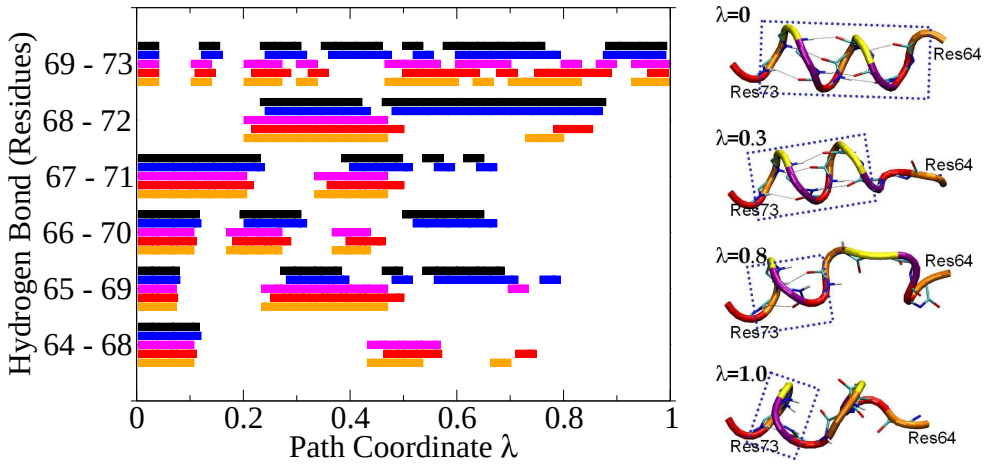


Figure 5.4: **A)** Evaluation of the α-helical hydrogen bonds present in Switch II along the 5 best transition pathways (See text for definition of present α-helical H-bonds). The structures along the 5 best transition pathways (black: best, orange: 5th-best path) were checked for such hydrogen bonds in residues $r = 64...69$ (Switch II). The horizontal axis measures the position along the path coordinate $\lambda = k/K$, where k counts the number of edges between the reactant and the current vertex, and K is the total number of edges in the path. Each colored line-segment means that a hydrogen bond is present at that point of the path of corresponding color. **B)** Structure of the Switch II helix (residue 64 to 73) along the best path. Residues that can form α-helical hydrogen bonds (black lines) are shown in equal colors. The actual α-helical part is denoted by a dotted box.

5.2.3 Structure of the Best-Path Network

What is the structure of the network consisting of the edges belonging to the best paths? This question is related to the question: what is the structure of the essential subspace, i.e. the set of configurations that are accessible at a given temperature. The best-path network is embedded in the essential subspace and "marks" the conformational transition routes which are, on average, most populated. For this, the 32 best paths with rate-limiting barriers within a range of 15 kcal/mol were computed. They describe a subnetwork of the full TN with $|\mathcal{V}_{BP}| = 180$ vertices and $|\mathcal{E}_{BP}| = 448$ edges.

A commonly used tool to reduce the dimensionality of a dataset to prepare it for visual representation is the Principal Component Analysis (PCA, [84]). In PCA, an Eigenbase is determined for the covariance matrix of the data. The principal components are the eigenvectors with the largest eigenvalues, *i.e.* those vectors in whose directions the variance of the data is maximal. If up-to-three eigenvectors account for the majority of the total variance of the data, it can be reasonably well visualized by projecting its coordinates on its up-to-three first principal components. For the vertices of the present best-path network 24 principal components are required to cover 95% of the variance in the data. A projection on three principal components therefore does not yield a projection, which captures the correct relative distances of vertices. Another projection method, Sammon mapping [85], which attempts to find an arrangement of low-dimensional data points such as to optimally reproduce the pairwise distances of the high-dimensional data, failed to achieve a small projection error because of the intrinsic high-dimensionality of the data. To cope with these difficulties, we developed another mapping method, the Topology-Preserving Mapping (TMP), which attempts to find an arrangement of low-dimensional data points such that the close data points (here: pairs of vertices which are directly connected by an edge) remain close, while all other data points must be further apart (but no specific requirements are made on their distance). It can thus be understood as a relaxed derivate of Sammon mapping. A detailed description of TMP is given in Appendix D.

A three-dimensional TMP projection of the best-path network is shown in Fig. 5.5. The 4 best pathways with rate-limiting barriers within 5 kcal/mol follow three largely disjoint channels: Apart from a junction close to the reactant that

is used by 3 paths, they explore different, non-adjacent regions of the essential subspace. This shows that despite a number of structural similarities shared by the best paths (such as the Tyr32 motion in Switch I and the H-bonding pattern in Switch II, see previous section), the precise conformations explored during the transition may vary significantly. This indicates that a large number of conformations can realize similar structural properties. Considering the 13 best paths within 10 kcal/mol, again significantly increases the variability of the transition. This subnetwork has more junctions: some of the higher-energy pathways deviate from lower-pathways just in a part of the way, rejoining them later. The full best-path network with rate-limiting barriers within 15 kcal/mol is a complex network that is particularly dense near the product structure, which can be accessed from many different conformations.

Fig. 5.6 highlights the emergence of dense regions in the best-path network by showing the best paths accessible at different levels of energy above the best path. At a level of 12 kcal/mol, there are a few conformational regions in which the best-path network is so dense, that these regions are likely to be fully accessible, only "interrupted" by some forbidden sub-regions, *e.g.* corresponding to steric clashes. This allows to switch between different pathways within those dense regions, *i.e.* they function as "hubs". Hubs are mutually connected by single pathways or narrow channels of pathways.

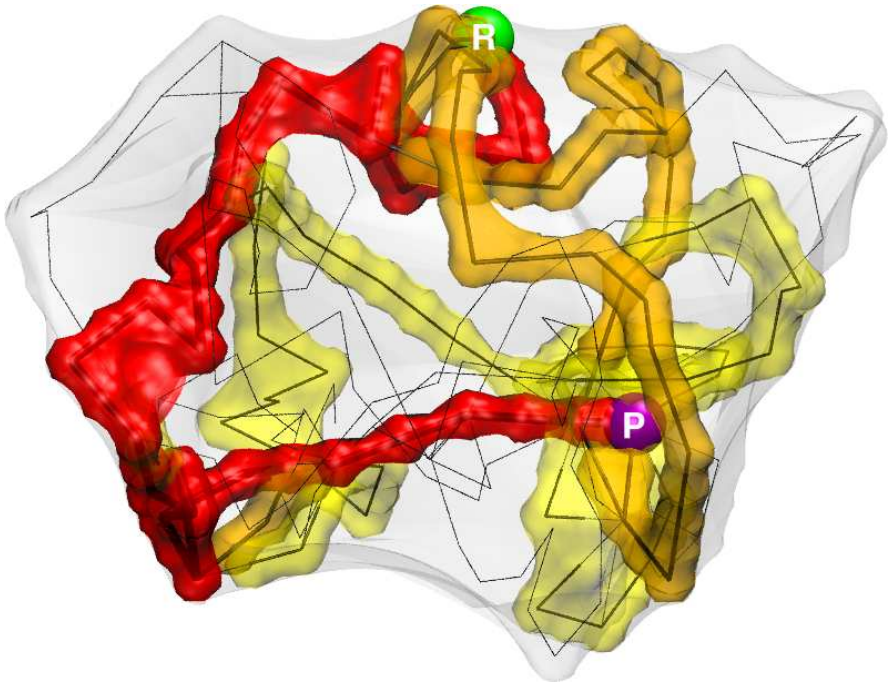


Figure 5.5: Three-dimensional projection of the TN vertices contained in the 32 best pathways between reactant (R) and product (P) whose rate-limiting barriers are in a range of 15 kcal/mol. Topology-Preserving Mapping (TMP, see Appendix D) was used for the projection. Each line junction or kink corresponds to a TN vertex, each straight line segment corresponds to a TN edge. The colored surfaces mark the regions explored by the best paths. Red: the best path, orange: 3 best paths with rate-limiting barriers within 5 kcal/mol, Yellow: 11 paths within 10 kcal/mol.

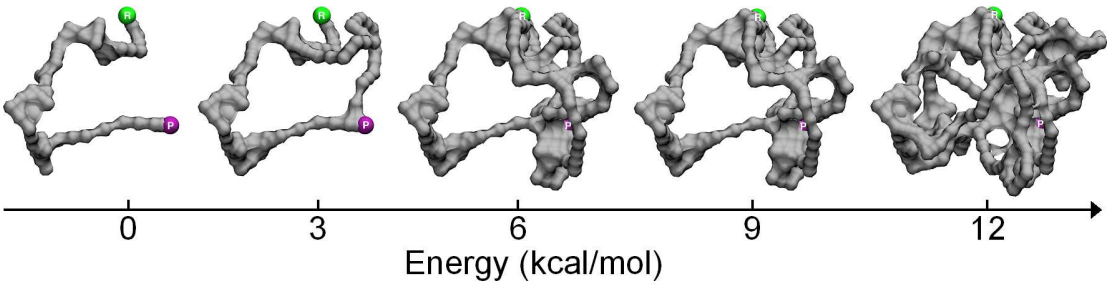


Figure 5.6: Accessible paths at different energy levels. The same projection as in Fig. 5.5 is used. Here, the best paths which are accessible at a certain energy (horizontal axis, relative to the rate-limiting barrier of the globally best path) are shown.

5.3 Energy Ridges and Rate-limiting Steps

To better characterize the mechanism of the rate-limiting steps of the transition, the two energy ridges of the Switch I and Switch II rearrangements were determined. The Switch I energy ridge (abbreviated as: ridge 1) is due to the rearrangement of Tyr32 as it passes from a conformational region with $-30° < \alpha < -10°$ to a region with $60° < \alpha < 110°$ (where α characterizes the position of Tyr32 and is measured as the dihedral angle P_β,C_{32},N_{32},OH_{32}). The other energy ridge (ridge 2) is the globally-highest energy ridge and is due to rearrangements in the Switch II.

5.3.1 Computational Effort

Ridge 1 was computed with $\Delta E_{\text{sure}} = 30$ kcal/mol and using *a priori* values for the barrier bounds ($B_{uv}^{\min} = 0$, $B_{uv}^{\max} = \infty$). Ridge 2 was computed fully (*i.e.* $\Delta E_{\text{sure}} = \infty$), with both *a priori* barrier bounds and with statistical estimates for the barrier bounds (see Appendix C.3).

To test the performance of the algorithm given in Sec. 3.3, the number of energy barriers needed to be computed by CPR to determine Ridge 2 was evaluated using different settings. The results are shown in Table 5.2, where the counting is started from scratch for each setting, assuming that no energy barrier has been computed yet. When the full energy ridge is computed ($\Delta E_{\text{sure}} = \infty$), statistical estimates reduce the computational cost by a factor of less than 2. This agrees with the results on random TN in Sec. 3.4.3, where it was shown that using statistical estimates to compute energy ridges gives less computational savings than for best paths. However, when only the lowest energy barrier of the ridge is determined ($\Delta E_{\text{sure}} = 0$), a factor of 4 is saved when statistical estimates are used instead of *a priori* barrier bounds. The statistically determined energy ridge 2 ($\Delta E_{\text{sure}} = \infty$) agrees with the "exact" ridge 2 in the lower-energy edges (up to 5 kcal/mol above the lowest edge). For edges of higher energy, only about 25% of the edges in the "estimated" ridge really belong to the true energy ridge. This confirms that the use of estimated barrier bounds is less safe for the determination of energy ridges than of best paths (see Sec. 3.4.3).

ΔE_{sure}[d]	Prior[d]	Est.[d]
0	805	214
5	862	293
10	897	383
15	1092	509
20	1092	622
∞[c]	1092	667
Size[a]	174	162
Barr.[b]	45.7	45.7

Table 5.2: Number of edges that must be computed with CPR to determine the energy ridge of the Switch II rearrangement, assuming no energy barrier has been previously computed. a) Number of edges in the fully determined energy ridge. b) Lowest edge barrier of the energy ridge, in kcal/mol relative to the reactant. c) ∞ means that all barriers of the ridge are determined. d) Same meaning as in table 5.1.

5.3.2 Structural Analysis

Energy ridge 1 and 2 impose an ordering of events on all possible pathways: The subnetwork defined by the reactant-side of ridge 2 is divided by ridge 1, and *vice versa* the subnetwork defined by the product-side of ridge 1 is divided by ridge 2. This implies that all trajectories from reactant to product must pass through ridge 1 before passing through ridge 2. Therefore, the rearrangement of Tyr32 is always finished before the rate-limiting rearrangement of Switch II starts. The position of the two energies ridges can be seen in Fig. 5.7.

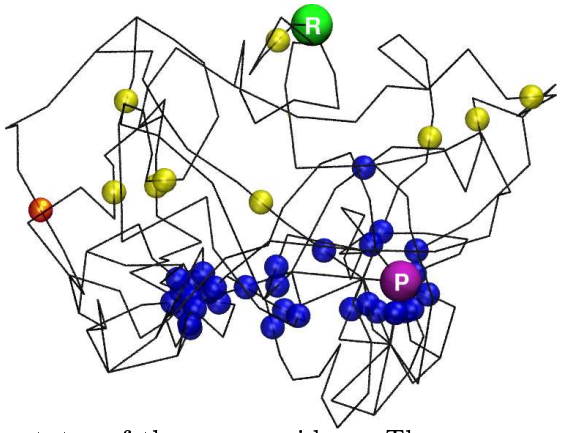


Figure 5.7: Transition states of the energy ridges. The same projection as in Fig. 5.5 is used. The edges whose transition states lie on an energy ridge are marked by colored balls. Yellow: energy ridge 1, Tyr32 is passing underneath the backbone. Orange: energy ridge 1, Tyr is passing through the solvent. Blue: energy ridge 2, which corresponds to the final rearrangement of Switch II towards the product conformation. The ridges mark an order of events in the Ras p21 conformational switch: Each pathway first passes through ridge 1 (*i.e.* the Tyr32 repositioning) before passing through ridge 2 (*i.e.* the rate-limiting Switch II rearrangement).

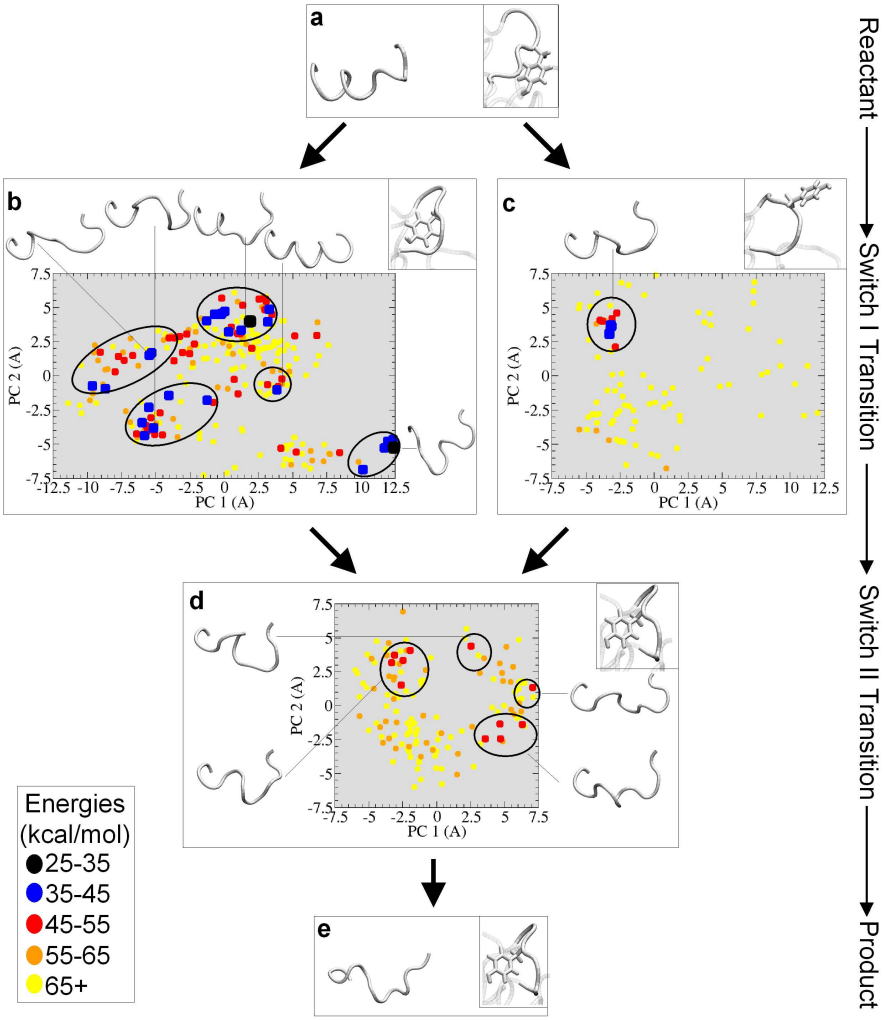


Figure 5.8: Two-dimensional projection of the energy ridges of the Ras p21 transition. Transition states from each ridge were projected on their two first Principal Components (computed from the C_α-coordinates). Each panel (b,c,d) shows one ridge and the corresponding conformation of the Switch I loop (box in top right corner of each panel). The projected points cluster (ellipsoids) according to their different Switch II conformations (typical backbone conformation shown for each cluster). The energy of the transition structures, is coded by color. (a) Reactant state: Switch I has Tyr32 pointing to the 'right', Switch II is a helix. From here, the conformational change proceeds through panels b or c. (b) Energy ridge of the Switch I-transition, with Tyr32 passing underneath the backbone. There is a large variety of alternative Switch II-conformations at this step of the transition. (c) Energy ridge of the Switch I-transition with Tyr32 moving through the solvent. (d) Energy ridge of the Switch II-transition, which is globally rate-limiting. The transition of Switch I is already completed and Tyr32 is pointing to the 'left'. Various iso-energetic ways for the Switch II rearrangement coexist. (e) Product state: Switch I is pointing to the 'left' and Switch II helix has fully unfolded.

To visualize the two energy ridges, Fig. 5.8 shows a two-dimensional projection of the transition states contained in ridge 1 and ridge 2. Ridge 1 was split into two sets: one set containing the transition states that involve the passage of Tyr32 underneath the backbone, and the other set containing the transition states having Tyr32 passing through the solvent. In the case where passage of Tyr32 is underneath the backbone, there are 7 different transition states in Ridge 1 up to 10 kcal/mol above the lowest transition state in Ridge 1. These considerably differ in the amount of unfolding of the Switch II helix: some still form a perfect helix, while in others the helix is fully unfolded (Fig. 5.8b). In the unlikely case that Tyr32 passes through the solvent, the conformation of the partially unfolded Switch II helix is well defined, as can be seen from its similar structure in the next-higher transition states (Fig. 5.8c).

Ridge 2 contains the globally rate-limiting transition states. 14 of them are up to 10 kcal/mol above the lowest transition state in ridge 2 (which is the rate-limiting barrier in the globally best path). These alternative transition states are highly scattered in Fig. 5.8d, showing that the structure of Switch II varies considerably. Thus, there are many different ways in which Switch II can rearrange toward the product structure (in agreement with Sec. 5.2.3) and the coupling between Switch I and II is weak enough to allow for different orders of the conformational events in both Switch regions. This confirms that the Ras p21 conformational switch is highly degenerate, thus involving a possibly significant entropic contribution to the free energy profile of the conformational switch [19].

5.4 Energy Distribution in Transition States

To further extend our understanding of the structure and interactions in the transition states, an energy decomposition was obtained for all saddle points of both energy ridges. The concept here is to decompose the potential energy of a given structure, $E_{\mathrm{pot}}(\mathbf{x})$, into contributions by the individual residues, $E_{\mathrm{pot},R}(\mathbf{x})$:

$$\begin{aligned} E_{\mathrm{pot}}(\mathbf{x}) &= \sum_{\text{pairs}\,(i,j)} e_{i,j}(\mathbf{x}_i,\mathbf{x}_j) = \sum_{\text{res}\,R}\left[\sum_{(i,j)\in R} e_{i,j}(\mathbf{x}_i,\mathbf{x}_j) + \frac{1}{2}\sum_{i\in R, j\notin R} e_{i,j}(\mathbf{x}_i,\mathbf{x}_j)\right] \\ &= \sum_{\text{res}\,R} E_{\mathrm{pot},R}(\mathbf{x}). \end{aligned} \tag{5.1}$$

The contribution by each residue is the sum of its self-energy (which is the sum over all interactions internal to that residue[3]) and half of the interaction energy with the rest of the protein (which is the sum over all interactions between atoms of that residue and atoms of other residues). Such a decomposition may yield a detailed understanding of the transition state and its energetics which are given by the interactions of the individual residues.

5.4.1 Energy Decomposition for Lowest Saddle Points

Here, the energies of all saddle points in energy ridge 1 and 2 were decomposed into the contributions of the Switch I residues, the Switch II residues, the magnesium and GDP ligands and four water molecules which act as magnesium coordination partners. Fig. 5.9 shows a visualization of the Switch I residue energies for the lowest-energy transition states in energy ridge 1. There are several ways how the rearrangement of Tyr32 can occur in detail. In the reactant state, the flexibility of Switch I is limited as the Thr35 backbone is attached to the magnesium, which hinders the passage of Tyr32 underneath the backbone. To enable the Tyr32 passage, there are three options:

1. Switch I is deformed such that Tyr32 can avoid Thr35 (Fig. 5.9B and C). The residue energies are relatively equally distributed over Switch I.

2. Thr35 temporarily detaches from the magnesium, opening a cleft through which Tyr32 passes (Fig. 5.9A and D). Thr35 and the Mg ion are energetically excited. The lowest-energy transition state shown in Fig. 5.9A is more beneficial than the one shown in Fig. 5.9D because Thr35 is stabilized by a hydrogen bond to the backbone of Asp33.

3. Thr35 detaches from the Mg ion and is replaced by Water173 (Fig. 5.9E). Switch I now has enough flexibility to let Tyr32 pass. The energies of both Water173 and the magnesium are reduced because of their newly-formed interaction. The GDP energy increases as its relative contribution to the coordination of the Mg ion is reduced. The Thr35 energy increases because it has lost the Mg ion as a coordination partner.

[3]If an implicit water model is used, as is the case here, the residual self-energy also contains contributions from solvation and hydrophobic energies.

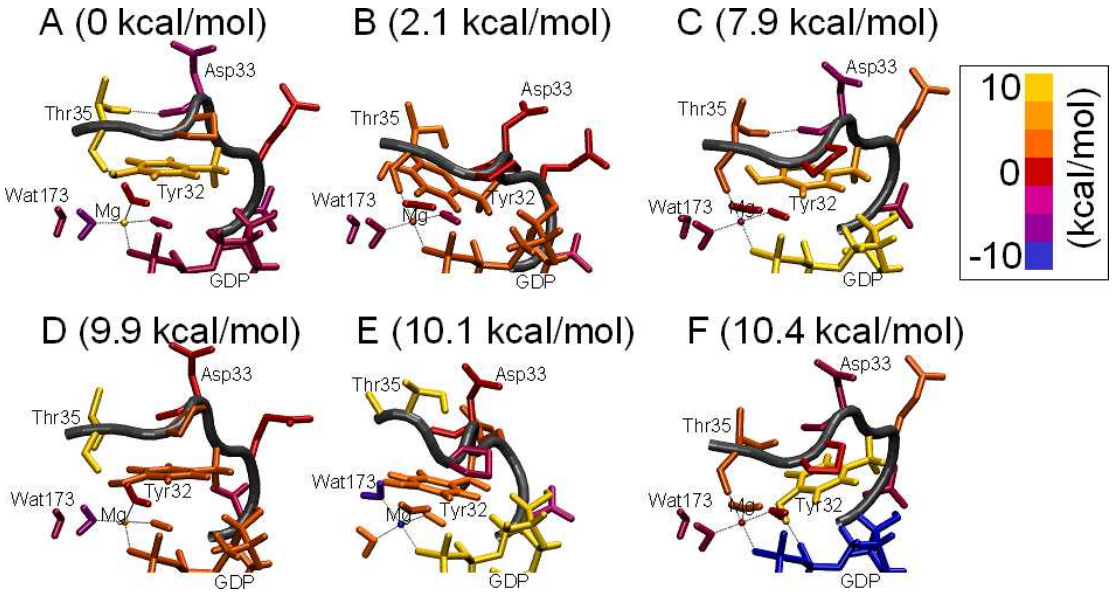


Figure 5.9: Visualization of the residue energies for the best six transition structures in ridge 1. The total energies, relative to the best transition structure (A) are given in parentheses. The residue energies, relative to their average value in the reactant and product end-states, are computed as defined in Eq. 5.1 and coded by color (see legend).

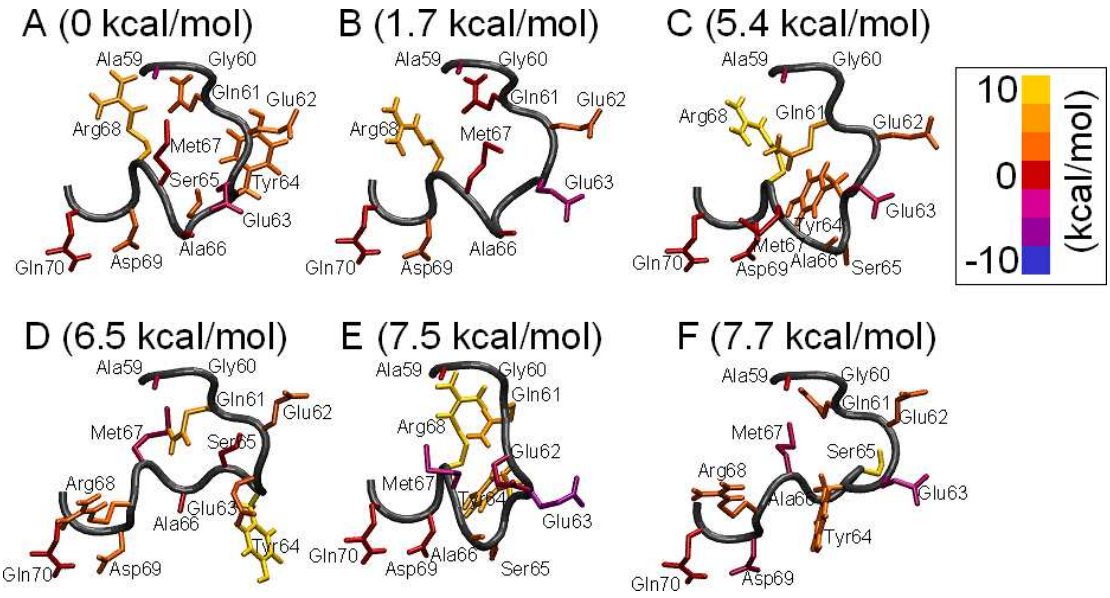


Figure 5.10: Visualization of the residue energies for the best six transition structures in ridge 2. See also Fig. 5.9.

The energy decomposition of the ridge 2 structures into Switch II residues shows that the large amount of variation in ridge 2 is not only in the structures but also in the residue-energy patterns (see Fig. 5.10). The residues at the Switch II boundary, Ala59, Gly60, Asp69 and Gln70 have similar conformations and in the different transition state energies. The conformations of the intermediate residues Gln61 to Arg68 differ significantly as well as the distribution of energies. Arg68 is (at least slightly) excited in all cases. Tyr64 is considerably excited in Fig. 5.10D and E, Gln61 in Fig. 5.10D, and Ser65 in Fig. 5.10F.

5.4.2 Correlations between Residue Energies

A more comprehensive picture of the interactions between individual residues is obtained when correlations between the residue energies are analyzed. In energy ridge 1, for example, we expect a negative correlation between the energies of Wat173 and Thr35 as they are competing for interaction with the magnesium ion (path 3, above). The energies of Tyr32 and Asp33, however, should be positively correlated as they collaborate to facilitate the Tyr32 transition (path 2, above).

The correlation $c_{ij} \in [-1, 1]$ between two residue energies E_i and E_j is computed as:

$$c_{ij} = \frac{\langle E_i E_j \rangle - \langle E_i \rangle \langle E_j \rangle}{\sigma(E_i)\sigma(E_j)}, \tag{5.2}$$

where $\langle E_i E_j \rangle - \langle E_i \rangle \langle E_j \rangle$ is the covariance between E_i and E_j, weighted by their standard deviations $\sigma(E_i)$ and $\sigma(E_j)$. By computing the correlations between the residue energies over all transition states in ridge 1 and ridge 2 each, two correlation matrices are obtained. The results are shown in Fig. 5.11.

In ridge 1 (Fig. 5.11A and B) the strongest correlations are within the ligand cluster and Switch I loop. There are strong (anti)correlations between the magnesium ion and its coordination partners. These correlations are of little interest as the coordination sphere does not undergo conformational changes. The only exception here is, the Wat173 whose energy is correlated with the magnesium energy and anti-correlated with the energies of the magnesium coordination partners. This is because Wat173 is separated from the magnesium in the reactant state, but within the coordination sphere in the product state, thus competing

with GDP and the other waters.

Among the Switch I residues, there is an anti-correlation between the energy of Thr35 and the energies of Tyr32 and Asp33. This is explained by different competing pathways for the Tyr32 passage. If the Tyr32 passage is facilitated by detaching Thr35 from the magnesium (path 2, above), the Tyr32 energy is comparatively low as it can pass unhindered, and during the passage the hydrophobic ring is buried and the OH-group can interact with Thr35 and the magnesium. The Asp33 energy may also decrease as its backbone can form a strong hydrogen bond with Thr35. However, the energy of Thr35 is high as the very favorable interaction with the magnesium ion is lost. In the alternative pathway (path 1, above), Tyr32 is threaded underneath the backbone by a deformation of Switch I. This pathway is less favorable for Tyr32 because of steric clashes and for Asp33 as it cannot hydrogen-bond with Thr35. It is very favorable for Thr35 as it can remain in the coordination sphere of the magnesium ion. The third possible pathway (path 3, above) involves Wat173 which replaces Thr35 in the magnesium coordination sphere. As expected, there is an anti-correlation between the Wat173 and Thr35 energies (reflecting the competition of these residues for the magnesium ion coordination) while Wat173 is positively correlated with Tyr32 and Asp33 (as these residues do not need to be deformed in path 3).

The main correlations in the energy ridge 1 transition states are among Switch I and the ligands while the correlations with Switch II and within Switch II are relatively weak. The possible transition states are structurally similar, but there is significant competition between the residues for three pathways of moving Tyr32 underneath the backbone that are different in detail. This competition leads to a *frustration* between the participating residues within the ridge (*i.e.* there is no "perfect" transition state configuration in which all residue energies are low), reflected by anti-correlations between these residues' energies.

The picture of the transition over energy ridge 2 (Fig. 5.11C and D) is considerably different. Apart from its boundary residues 59,60 and 69,70, the energetics and the structure of Switch II are not well-defined. Apart from some correlations internal to Switch I and the ligands there is a very clear block of positive correlations involving most residue pairs in Switch II. The energies of the residue triple Tyr64-Ser65-Ala66 are strongly correlated. Less strongly, but also positively correlated

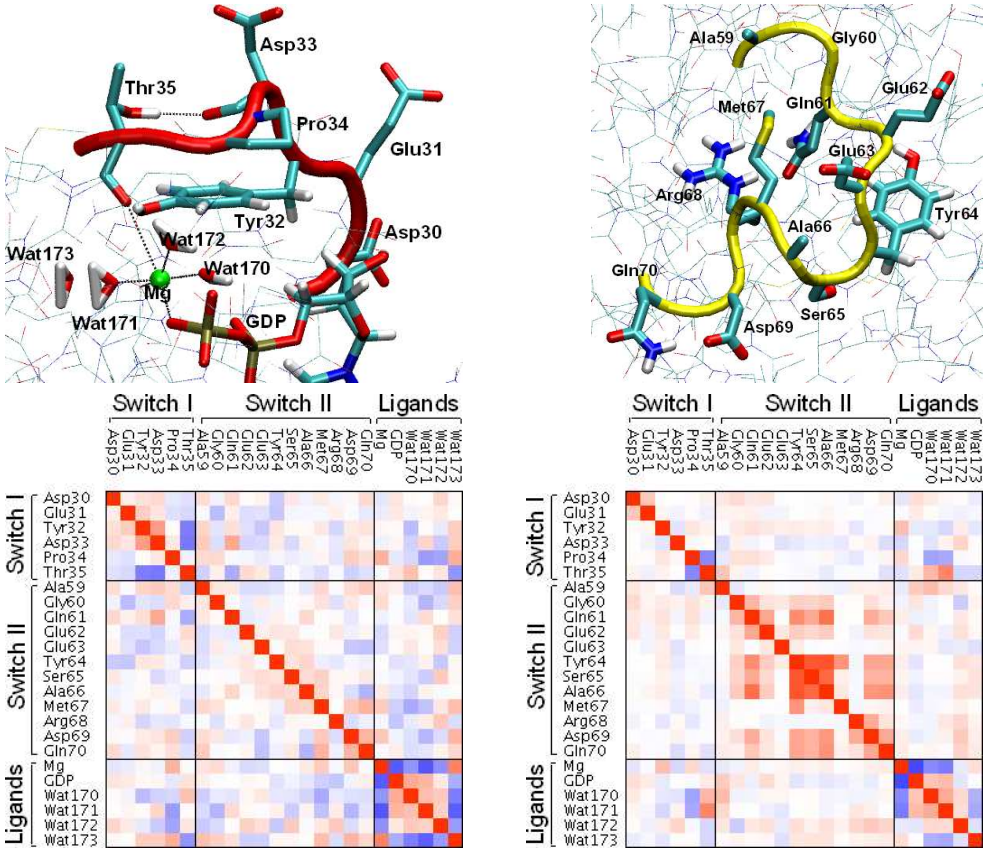


Figure 5.11: Relationship between transition state structure and correlations in between residue energies. A) Best transition state structure in energy ridge 1. B) Cross-correlations between residue energies, computed over all transition states of energy ridge 1. Anti-correlations are shown in blue, positive correlations in red. C) and D) are same as A) and B), but for energy ridge 2.

are the triples Gly60-Gln61-Glu62 and Arg68-Asp69-Gln70, as well all of these three triples with their neighbor triples. No significant anti-correlation is present between the Switch II residue energies.

What is the impact of such a cluster of positive correlations on the total energy of the system? To answer this question, imagine following model system: We are given a number n of degrees of freedom in a heat bath with temperature T. Each degree of freedom, i, can assume either of two states 0 and 1 with different potential energies $E_i \in \{0\, k_B T, 1\, k_B T\}$. If the degrees of freedom are independent, each degree of freedom will have its state population given by the Boltzmann distribution ($p(0) = 1/(1 + \mathrm{e}^{-1})$, $p(1) = \mathrm{e}^{-1}/(1 + \mathrm{e}^{-1})$). The total

potential energy of the system is given by $E_{\text{pot}} = \sum_i E_i$. Thus, the total potential energy is distributed as

$$p(E_{\text{pot}}) = \binom{E_{\text{pot}}}{n} p(0)^{n-E_{\text{pot}}} p(1)^{E_{\text{pot}}}, \tag{5.3}$$

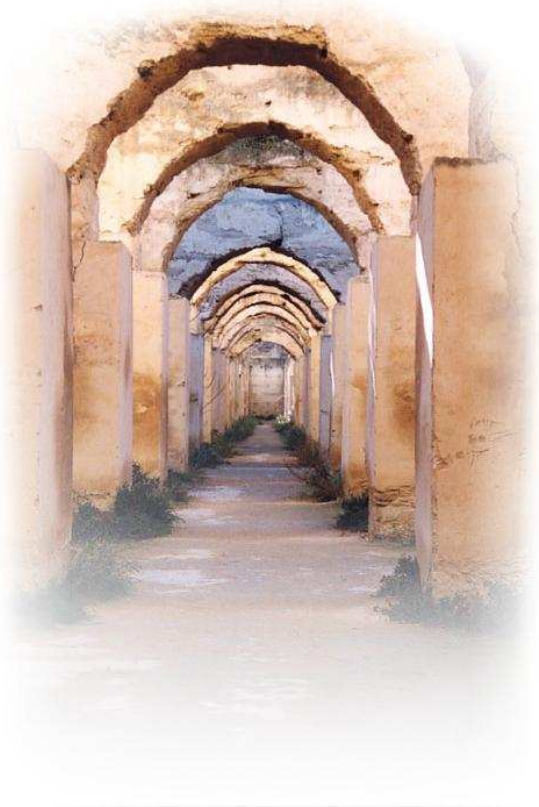
i.e. as a Poisson distribution.

If there are correlations between the energies of the individual degrees of freedom, the total energy distribution will deviate from Eq. (5.3). Anti-correlations between individual degrees of freedom sharpen the energy distribution because high values in some degrees are compensated by low values in others. Positive correlations, on the other hand broaden this distribution as the system prefers to collectively explore both low and high-energy states. Thus, a pairwise positive correlation between residue energies, as is visible for the Switch II in Ras p21, above, increases the likelihood that this part of the protein explores extreme potential energy values. It is therefore able to faster overcome high potential energy barriers, such as the one involved in unfolding the Switch II Helix. Note that this effect is neither enthalpic nor entropic: In fact it does not change the energies at all, but rather the magnitude of excitations on certain parts of the energy landscape.

It may be a general evolutionary strategy to craft parts of proteins in such a way that collaborative inter-residual correlations that are active in particular conformations allow to assume otherwise inaccessible states.

Chapter 6

Conclusion and Outlook



6.1 Conclusion

This is a computational work with a molecular biophysical application. In the molecular biophysics community, problems are typically not formalized in such a way that they can be addressed by a computer scientist, whereas the computer science community does not yet seem to pay much attention to the field of biophysics. For example, some basic ideas of Transition Networks have been formulated [35, 50], but no rigorous computational formalization had been made. Except for some very recent work [86, 87], no graph-theoretical methods had been used to address this class of problems. In the present work, such a formalization was laid down. We feel this is an important step in providing a basis of communication between the fields of computer science and biophysics, at least in the problem field of conformational transitions.

Hitherto it was not possible to generate and analyze the vast Transition Networks involved in complex protein transitions for two reasons: (1) no efficient sampling procedure was available to find and distribute the network vertices and (2) the calculations to determine the energy barriers of the many edges are expensive. Both these problems have been overcome in the present study. An efficient sampling, selection and minimization procedure generates uniformly distributed conformers in a conformational subspace that is energetically accessible and geometrically relevant for the transition. The presented graph-theoretical approaches allow to determine global properties of the network, such as best paths connecting and the energy ridges separating the end-states while computing only a small subset of the total number of sub-transitions in the network.

When applied to the conformational switch of Ras p21, the energetically best pathways and the two main energy ridges could be identified. These results give detailed information on the structural mechanism of the conformational switch which had previously been subject to speculation [19, 81, 82]: 1) The rearrangement of Switch I always occurs such that Tyr32 is threaded underneath the protein backbone. Within this restriction, there exist at least three competing pathways: a) the Thr35 detaches from the magnesium ion coordination sphere such as to open a cleft for the Tyr32 passage, b) Wat173 replaces Thr35 in the magnesium ion coordination sphere, allowing Tyr32 to pass freely, or c) Switch I is deformed

such that Tyr32 passes while avoiding the magnesium coordination sphere. 2) The rearrangement of Switch I is finished before the rate-limiting rearrangement of Switch II starts. 3) The hydrogen-bonding pattern of the Switch II helix unfolding is similar in the best paths. The general tendency is that the helix unfolds from the nucleotide-near (N-terminal) to the nucleotide-far (C-terminal) end. 4) Despite the above similarities, the precise order of conformational events in Switch I and II and the detailed way of rearrangement in Switch II varies substantially. This shows that complex conformational transitions in proteins such as Ras may occur *via* multiple pathways.

An analysis of the transition-state energetics for contributions by the individual residues showed that there is considerable competition between the Switch I residues, supporting the view that a few mutually exclusive pathways for the Switch I rearrangement exist. In contrast, the residue energies in Switch II are cooperative, thereby enhancing the ability to overcome large potential energy barriers which promotes the unfolding of the Switch II helix.

As the Ras p21 application demonstrates, the methodology developed here is useful to understand very complex mechanisms in proteins independent of their typical timescale. This was hitherto impossible and represents a significant methodological progress in the field of molecular biophysics.

6.2 Outlook

This work has inspired a number of promising follow-up projects concerning the extended application of the methods presented, further methodological development, and the fundamental understanding of proteins and other complex systems.

The methodology developed here is applicable to complex conformational changes in many proteins whose functional timescale and complexity precludes the use of direct simulation. An application that is related to work on Ras p21 here would be to compute Transition Networks for the conformational switch of other members of the GTPase protein super-family, such as Ran and Rap [88]. These proteins have different functions than Ras p21, but are speculated to have a similar mechanism as they share some structural properties and they also involve a conformational change that is triggered by the hydrolysis of GTP. Their mechanism and simi-

larities with the Ras p21 mechanism could be explained with Transition Network analyses. Using the methodological tools from this work, this would only require an investment of several weeks of CPU time. Any *general* findings on the mechanism of GTPases is not only interesting to a broad audience, but would also enlighten our understanding of a large number of structural-biological processes in the cell [89].

A major limitation of the results obtained from the Ras p21 Transition Network so far is that they are qualitative rather than quantitative in nature. This is because the Transition Network weights are given by potential energies, rather than free energies. The reliable computation of free energies for complex systems such as proteins is an open problem of great interest in the biophysical chemistry community [55]. Its difficulties are given by the sampling problem: "Physical" sampling methods, like molecular dynamics or Monte-Carlo, are in principle a rigorous approach to obtain free energies, but can in practice not achieve convergence for free energy values as they fail to sample a sufficient volume of the conformational space in the limited simulation time. The main obstacles here are a) high energy barriers which are not overcome, and b) large-scale diffusional motions which are not fully explored. Both problems could be addressed by combining these physical sampling methods with the "computational" sampling approach presented in Chapter 4. Problem a) may be avoided by distributing a number of samplers across the accessible conformational subspace according to the sampling method proposed here. Problem b) may be overcome by defining partitions of the conformational space with the aid of the sampled points, such that each sampler only has to explore a limited region of the diffusive motion.

A third project involves research on the physical properties underlying the dynamics of proteins as complex systems. A concept that has recently received considerable attention is that of the essential subspace of a protein, or generally of a complex dynamical system [15]. The essential subspace is the part of conformational space that is accessible to the protein at a given temperature. Available knowledge on the form, size and connectivity of that space has been obtained either indirectly from interpreting experimental data, such as relaxation times [90], or by analysis of simulation data [91], the latter of which is of course limited by the sampling problem. The Transition Network presented here samples the full conformational subspace that is relevant for a given transition, so our data

is an ideal basis for analyses of large-scale properties of the essential subspace: What is the general *form* of this space, *i.e.* is it compact, star-like, or more like a labyrinth? Is there any small-world behavior, *i.e.* is there a short conformational route between any pair of accessible conformations? What is the connectivity of the space, *i.e.* are there regions which are more dense than others and is this related to functional importance? Is there any relationship between the energy of a region and the number of neighboring regions? Does the connectivity of the essential subspace have any impact on the dynamics of the system (*e.g.* through entropic effects)? Answering these questions might help us not only to understand the physics of proteins but also to enhance our ability to model and predict complex system dynamics in general.

Appendix A

Algorithmic proofs

A.1 Energy Ridge

Here we prove the correctness and the time complexity of the energy-ridge algorithm (Algorithm 1, p. 34)

Correctness: We require that the minimum edge energy of the energy ridge is higher than the end-state energies, *i.e.* that there is a real barrier. Furthermore the Transition Network is required to have a pathway between the transition endstates. Given these conditions, Algorithm 1 returns the optimal energy ridge for the transition network.

1. At the end of the algorithm, all edges are i_R-i_P, i_R-i_R or i_P-i_P edges:

   Whenever a set of vertices V_1 that belongs neither to i_R nor i_P is connected to a set of vertices V_2that belongs to i_R or i_P, V_1 is also changed to i_R or i_P (step 3.3). Vertices which belong to i_P or i_P are never put into another group. Since the Graph is connected, all vertices will be i_R or i_P vertices and therefore all edges will be i_R-i_P, i_R-i_R or i_P-i_P.

2. The algorithm returns the set of i_R-i_P edges: The only step where a i_R-i_P edge is created is when an i_R-vertex and an i_P-vertex are connected (in 3.1) and then this edge is added to the set ER. A different edge can never become a i_R-i_P edge, as classes i_R and i_P are not added after the initialization and a

i_R-X or i_P-X would be transformed to i_R-i_R or i_P-i_P immediately. Therefore, ER contains all i_R-i_P edges and ER is returned from the algorithm.

3. The set of i_R-i_P edges is a cut: It separates the i_R set which is connected to the reactant from the i_P set which is connected to the product.

4. The energy ridge consists of i_R-i_P edges: Consider each edge e in the energy ridge in the order of ascending edge energies. For each e, there is at least one path from either vertex of e to either endstate (say v_R to e_0 and v_P to e_1) whose maximum edge energy is not higher than E_e. Therefore, e_0 has been connected to i_R and e_1 to i_P before e is considered and e is a i_R-i_P edge.

5. There are not other i_R-i_P edges in the network: Since the energy ridge is a cut (see 3) and all edges of the energy ridge are i_R-i_P edges (see 4), there can be no other i_R-i_P edges, otherwise the set of i_R-i_P edges would not form a cut (and therefore produce a contradiction with c).

Combining 1)-5), it is proved that the algorithm returns the set of edges comprising the energy ridge. ■

Complexity

The complexity of the energy-ridge algorithm is at most $O(|\mathcal{E}|\log|\mathcal{E}| + |\mathcal{V}|\log|\mathcal{V}|)$.

The edge sorting in step (1) is done with a standard sorting algorithm (*e.g.* quicksort), which has a complexity of $O(|\mathcal{E}|\log|\mathcal{E}|)$ [92].

The loop in step (3) is iterated $|\mathcal{E}|$ times. If none of (3.2)-(3.3) is executed, this adds a complexity of $O(|\mathcal{E}|)$. If (3.2)-(3.3) are executed, additional computation time is spent to add vertices to the lists of other vertex-groups and to change their list indexes. To minimize the complexity, the smaller group is always added to the larger group (see Algorithm 3). We construct the worst case as follows: only pairs of groups with equal size are associated, during the whole iteration cycle. This may occur only if the number of associated vertices is a power of 2. To further maximize the size of vertex groups, we require that only a single vertex (one of the endstates) is on one side of the ridge, while all other vertices are on the other side. In the beginning, $(|\mathcal{V}| - 1)/2$ vertex-pairs are formed and for each pair one vertex needs to be reordered. As a last step, two equally-sized groups

are associated and $(|\mathcal{V}|-1)/2$ vertices need to be reorded. In general, a number of

$$\frac{|\mathcal{V}|-1}{2}\cdot 1+\frac{|\mathcal{V}|-1}{4}\cdot 2+\cdots+2\cdot\frac{|\mathcal{V}|-1}{4}+1\cdot\frac{|\mathcal{V}|-1}{2}=\log_2(|\mathcal{V}|-1)\frac{|\mathcal{V}|-1}{2}$$

vertices need to be reordered.

Therefore, a total complexity of $O(|\mathcal{E}|\log|\mathcal{E}|+|\mathcal{V}|\log|\mathcal{V}|)$ is obtained.

A.2 Efficient Computation of Best Paths

Termination: If the algorithm does not return in 2), one edge is refined in 3). The algorithm thus terminates after at most $|E|$ cycles.

Correctness: The returned path is the globally best path after termination:

Assume the algorithm returns P_{ret}, but the real best path is $P \neq P_{ret}$.

If all edge energies of P are determined, P is computed as best path in 1), then it must be $P \equiv BP^{\min} \equiv BP^{\max}$ in 2) and therefore $P \equiv P_{ret}$ (contradiction).

Otherwise, be $\{E_{1,2}, E_{2,3}...E_{k-1,k}\}$ the edge energies of the edges along P. Since $E_{ij}^{\min} \leq E_{ij}$, the path cost of $BP^{\min}$ in 1) was less or equal than that of P and therefore less than that of P_{ret} which means $BP^{\min} \neq P_{ret}$, so P_{ret} is not returned in 2) (contradiction). ■

Appendix B

Random Transition Networks

For the test of algorithms as those presented in Chapter 3, it is useful to have a model that generates random transition networks. Depending on the underlying physical system, Transition Networks can have various different topologies and edge weights. Nevertheless, Transition Networks are not well represented by purely random graphs [93] with random weights, as they do have the following particular properties:

Embedding

The TN vertices are embedded in a D-dimensional space. Typically, the degrees of freedom of the system are strongly coupled, so that there is a strong correlation amongst them. As a result, the system configurations mostly reside in an essential subspace with a dimensionality much lower than D. It has been shown that for proteins with many thousand degrees of freedom less than 1% of the degrees of freedom is sufficient to cover most of the variance in the data [15]. Therefore, the random TN vertices are here embedded in a ($C \ll D$)-dimensional space. For the vertices of all random TNs in this study we choose a $C = 5$-dimensional hypercubic space. Molecular conformations are often specified in terms of torsion angles. To achieve a similar coordinate system, each dimension of the random TN embedding space has values in $]-180, 180]$ degrees and periodic boundary conditions. Initially, all $|V|$ vertices are embedded as random points in this space.

Connectivity

Given the positions of the vertices, the connectivity of the network is defined by drawing an edge between all vertex-pairs within a distance of d (Here, a root mean square distance of 40° is used). A commonly used measure for the connectivity of the network is its degree distribution $p(k)$: The degree of a vertex, k, is the number of neighbors it has; the degree distribution is the probability distribution of all vertex degrees in the network. In TN, the degree of a vertex is equal to the number of other vertices within a hypersphere of radius d, so that the degree distribution also is a measure for the vertex density distribution. For a random TN to be representative, it must have a degree distribution that is typical for the class of physical systems one is interested in.

For TNs of molecular conformational changes, we have found that the degree distribution is of a poisson type. The degree distribution of the Ras p21 TN analyzed in Chapter 5 is shown in Fig. B.1. All random TNs in this study are generated in such a way that they exhibit the same degree distribution.

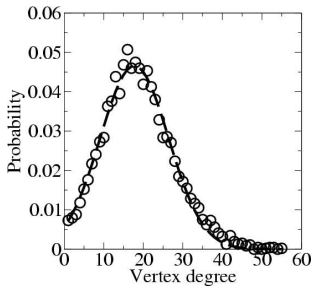


Figure B.1: Degree distribution of the Ras p21 network. The distribution is Poisson-like, closely following a Gaussian distribution around the mean of 19 (dashed line).

To obtain a random network with a predefined degree distribution $p_{\text{ref}}(k)$, we use following Monte-Carlo algorithm on the initial vertex embedding:

Algorithm 5 Random TN Embedding

(1) Given an initial vertex embedding $(\mathbf{x}_1, ..., \mathbf{x}_{|V|})$, compute the adjacency list and from this the degree distribution $p(k)$. Compute the distribution error $\epsilon_{p(k)} = \sum_{k=0}^{\infty}(p(k) - p_{\text{ref}}(k))^2$

(2) $i := 0$. While $i < i_{\max}$ and $\epsilon_{p(k)} > \epsilon_{\text{tol}}$, repeat:

(2.1) Randomly chose a vertex v with embedding $\mathbf{x}_v$ and randomly choose a new point $\mathbf{x}'_v$ in the embedding space.

(2.2) Compute the degree distribution $p(k)'$ for the case that v is moved to $\mathbf{x}'_v$ and the distribution error $\epsilon_{p(k)'}$].

(2.3) If $\epsilon_{p(k)'} < \epsilon_{p(k)}$, accept move: $\mathbf{x}_v := \mathbf{x}'_v$, $p(k) := p(k)'$, $\epsilon_{p(k)} := \epsilon_{p(k)'}$

The algorithm terminates when the maximum allowed error in the distribution, ϵ_{tol}, or the maximun number of iterations, $i_{\max}$, is reached.

TN weights

In TNs, the weights correspond to the form of the energy surface of the underlying physical system. Two extreme cases are (a) the energy surface has the form of one large basin (with some local roughness) and (b) the energy surface has no underlying form, the weights are just uncorrelated random numbers. Between these two extremes are the cases of local structure (c): the energy surface has a number of basins (with some local roughness), which are mutually connected. We propose a multi-harmonic-basin model as given by the following algorithm:

Algorithm 6 Random Transition Network

(1) Generate Topology according to $p_{\text{ref}}(k)$
(2) Place $n_S = 1/o$ seeds randomly on different vertices
(3) For each vertex v:
  (3.1) $d =$ distance to next seed
  (3.2) $E_v = d^2 + \text{Gaussian}(0, \sigma)$
(4) For each edge $e = (u, v)$:
  (4.1) $E_e = \max\{E_u, E_v\} + |\text{Gaussian}(0, \sigma)|$

Here, Gaussian(μ, σ) generates a random value drawn from a Gaussian distribution with mean μ and standard deviation σ. The parameter n_S is equal to the number of harmonic basins underlying the energy function, the order-parameter o, which is the inverse of n_S, quantifies the amount of order on the potential energy surface. For $n_S = 1$ ($o = 1$), we have case (a) and for $n_S = |V|$ (o minimal), we have a *random noise network*, case (b). All values in between are examples for the case of local structure, case (c) (see Fig. 3.3). Unless stated otherwise, $\sigma = 1$ and $n_S = |V|$ are used in this study. Sec. 3.2.2 and 3.3.2 treats the effects of using different values of n_S.

In this study, best path(s) are computed on random TNs. This requires the definition of a pair of endstates for each computation. As transition endstates usually are at the energy minima of the endstate basins, the pair of endstates was not selected in a completely random way. Rather, two random vertices were chosen and then both of them were minimized on the network, *i.e.* each endstate was repeatedly moved to the lowest-energy neighbor vertex until all its neighbor

vertices had higher energies. If, after this minimization, both vertices coincided, the process was repeated until two distinct endstates were found.

Appendix C

Ras p21 Setup and Details

C.1 Ras p21 setup

Crystallographic structures exist for both the GTP-bound (Protein Data Bank structure 5p21 [78]) and GDP-bound states (1q21 [79]) of Ras p21. The γ-phosphate was deleted from 5p21, to yield the reactant state. The 1q21 structure served as product state. The HBUILD facility in CHARMM [8] was used to place the missing hydrogens.

All calculations were performed using the extended-carbon potential function (PARAM19) [94] and with version 2 of the Analytical Continuum Electrostatics (ACE) method to model the solvent [83]. Non-bonded interactions were smoothly brought to zero by multiplying them with a switching function between 8 and 12 Å.

The structure of a protein may be affected by the crystal environment. Therefore, both the reactant and the product structures were first relaxed using molecular dynamics simulations. For this 20 ps of heating were followed by 100 ps of equilibration and a 10 ns production run. One structure every 100 ps (making up 100 structures in total) was selected and energy minimized to a gradient RMS of 10^{-3} kcal mol^{-1} Å^{-1}. The structures with the lowest energies were selected as reactant and product structures. The potential energy of these structures was lower than that obtained by a direct minimization of 5p21 and 1q21 by 30-45 kcal mol^{-1}. Structurally, the differences compared to 5p21 and 1q21 were rather small, con-

sisting mainly of exposed side-chain rearrangements, while the backbone fold of the Switch regions was preserved. The RMS coordinate deviations from the directly minimized crystallograph end states were <1.8 Å for the non-fixed atoms (<2.4 Å for the switch regions).

To remove insignificant degrees of freedom, residues which were not involved in the conformational switch and whose atoms had similar positions in both end-states were fixed (residues 1-4, 42-53, 77-95, 110-115, 124-143, 155-167), leaving 1001 atoms free to move. To obtain the same positions for the fixed atoms in the two end-states, the product structure was oriented onto the reactant structure so as to minimize the RMS deviation between the fixed atom coordinate sets. Then, the reactant and product values of these coordinates were averaged. The averaged coordinates of the fixed atoms were used for all calculations. Furthermore, insignificant differences in the side-chains of non-Switch regions were removed from the end-states as described in [19]. Finally, both end-states were minimized to a gradient RMS of 10^{-3} kcal mol^{-1} Å^{-1}.

C.2 CPR SETUP

Here, the refinement algorithm used to determine edge energy barriers was Conjugate Peak Refinement (CPR, introduced in Sec. 2.1.1), which refines an initial path P to a Minimum Energy Path (MEP). Unless stated otherwise, the edge energies were determined in a single refinement step. All MEP were computed with the CPR code implemented in the TReK module of the CHARMM program [8] version 29, using the default CPR settings for identifying saddle points: With these settings, the gradient RMS at a saddle point is required to be smaller than $g_{\mathrm{sad}} = 0.05$ kcal mol^{-1} Å^{-1} for an uninterrupted number of $\sqrt{N}$ conjugate line minimizations, where N is the number of moving atoms (for Ras p21, $\sqrt{N} = 32$).

C.3 STATISTICAL ESTIMATION OF EDGE BARRIERS

A method is given for the statistical estimation of lower and upper bounds for the energy barriers of sub-transitions. For this, one correlates available information on the edges (u, v), such as distance between its vertices $\delta_{uv} = |\mathbf{x_u} - \mathbf{x_v}|$, with the

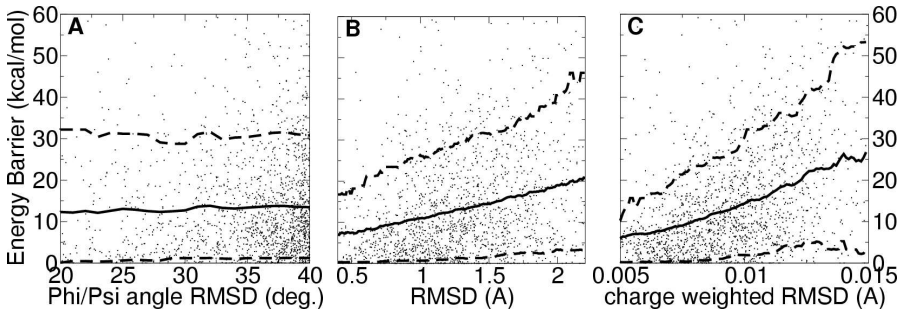


Figure C.1: Predicting lower and upper bounds to the edge energy barriers. The energy barrier is plotted versus the distance between the end-states of a given sub-transition in Ras p21, using different distance metrics: A) RMSD in ϕ/ψ-dihedral space of the **S**-regions, B) All-atom RMSD in Cartesian space C) same, but with each atomic distance weighted by the absolute atomic charge. 90% of the points lie below the upper dashed line, 10% below the lower dashed line. These were used as lower and upper estimates for the estimation of optimistic and pessimistic best paths. The solid line shows the average barrier.

already-computed energy barriers $B_{uv} = E_{uv} - \max\{E_u, E_v\}$. Using a certain confidence interval, one obtains upper and lower estimates, $B_{uv}^{min}(\delta_{uv})$ and $B_{uv}^{max}(\delta_{uv})$ which may be used to replace the strict edge-barrier bounds.

For Ras p21, after computing the first $\sim$ 2000 energy barriers, these barriers were correlated with the distance between the corresponding minima so as to yield a distance-dependent barrier estimate which was later used to replace the edge-weight bounds. Fig. C.1 shows plots of these barriers against three different distance measures. The average value and the boundaries of a 90% confidence interval are given. Clearly, the ϕ/ψ-RMSD is not a useful measure here as it is not correlated with the energy barrier. The Cartesian RMSD gives a better correlation whereas the charge-weighted RMSD, $d_C(\mathbf{x}, \mathbf{y})$, defined as:

$$d_C(\mathbf{x}, \mathbf{y}) = \sqrt{\frac{\sum_{i=1}^{N} (\mathbf{x}_i - \mathbf{y}_i)^2 q_i^2}{N}},$$

where N is the number of atoms and q_i is the charge on atom i, here gives the best correlation of the three distance measures. The 90% confidence interval bounds of the latter statistic were used as estimates for subsequent computations which used statistical estimates for the energy barriers.

Appendix D

Topology-Preserving Mapping (TMP)

Topology-Preserving Mapping (TMP) is a method that allows to map a manifold that is embedded in a high-dimensional space into a low-dimensional space for the sake of visualization.

If the manifold exploits many dimensions of the embedding space, it is not possible to find a projection that properly represents all pairwise distances between points of the manifold in a low-dimensional space. For this reason, mapping methods which consider all pairwise distances to be equally important, such as Principal Component Analysis [84] or Sammon's Mapping [85] fail to find an appropriate low-dimensional representation. TMP achieves a good low-dimensional projection by putting more emphasis on local connectivity (the topology) while it does not attempt to correctly reproduce large pairwise distances.

In particular, TMP distinguishes between (locally) connected and not (locally) connected points. TMP enforces that in the image space:

a) the distance of a pair of connected points lies within the range $[d_l, d_u]$. Thus, local connectivity must be represented in the image space.

b) the distance of a pair of unconnected points is larger than d_r. Typically, it is $d_r > d_u$, so than one can distinguish between connected and disconnected points in the image space.

One way to enforce these conditions is to define a cost-function

$$C = \sum_{u=0}^{|\mathcal{V}|-1} \sum_{v=u+1}^{|\mathcal{V}|} c_{uv}, \tag{D.1}$$

which sums over all pairs of points in the image space. Its terms are defined as:

$$c_{uv} = \begin{cases} (\min\{d_{uv}, d_l\} - d_l)^2 + (\max\{d_{uv}, d_u\} - d_u)^2, & \text{if (u, v) neighbors} \\ (\min\{d_{uv}, d_r\} - d_r)^2, & \text{otherwise.} \end{cases} \tag{D.2}$$

Here, d_{uv} is the distance of the points in the image space. To obtain a mapping, the points are first randomly distributed in the image space and varied in such a way as to minimize C.

TMP was used here to map the vertices of a Transition Network on a 3-dimensional space. Connectivity was deduced directly from the Transition Network: two vertices are locally connected if a direct edge exists between them and disconnected otherwise. To define the cost function, we set the reference distances to $d_l = 1.0$, $d_u = 1.2$ and $d_r = 2.0$ length units. The image points were restricted to be in a box of $40 \times 40 \times 40$ length units size. After distributing them randomly in the box, the cost function Eq. (D.1) was optimized using an adaptive-steplength Monte Carlo procedure. In each iteration, one random image point u was displaced:

$$\mathbf{x}'_u := \mathbf{x}_u + \lambda \mathbf{r}.$$

Here, $\mathbf{x}$ and $\mathbf{x}'$ are the old and the new positions of point u, $\mathbf{r}$ is a random vector whose coordinates are drawn from a uniform distribution over $[-1, 1]$, and λ is the steplength (initially 40). A move was only accepted if $\mathbf{x}'_u$ was within the box and if this step reduced the value of C. If 100 steps were made without any acceptance, λ was reduced by multiplying it with a factor of 0.75, but was never reduced below 0.2 length units. The optimization was stopped when no move was accepted during the previous 10^5 steps, which happened after about $5 \cdot 10^5$ steps in total.

Bibliography

[1] M. F. Perutz, A. J. Wilkinson, M. Paoli, and G. G. Dodson. The stereochemical mechanism of the cooperative effects in hemoglobin revisited. *Annu. Rev. Biophys. Biomol. Struct.*, 27:1–34, 1999.

[2] K. Olsen, S. Fischer, and M. Karplus. A continuous path for the $T \rightarrow R$ allosteric transition of hemoglobin. *Biophys. J.*, 78:394A, 2000.

[3] M. A. Geeves and K. C. Holmes. Structural mechanism of muscle contraction. *Annu. Rev. Biochem.*, 68:687–728, 1999.

[4] S. Fischer, B. Windshuegel, D. Horak, K. C. Holmes, and J. C. Smith. Structural mechanism of the recovery stroke in the myosin molecular motor. *Proc. Natl. Acad. Sci. USA*, 102:6873–6878, 2005.

[5] M. L. Coleman, C. J. Marshall, and M. F. Olson. Ras and Rho GTPases in G1-Phase Cell-Cycle Regulation. *Nat. Rev. Mol. Cell Bio.*, 62:851–891, 1993.

[6] A. B. Vojtek and C. J. Der. Increasing Complexity of the Ras Signaling Pathway. *J. Biol. Chem.*, 32:19925–19928, 1998.

[7] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucl. Acids Res.*, 28:235–242, 2000.

[8] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus. Charmm: A program for macromolecular energy, minimization, and dynamics calculations. *J. Comp. Chem.*, 4:187–217, 1983.

[9] D. A. Pearlman, D. A. Case, J. W. Caldwell, W. R. Ross, T. E. Cheatham III, S. DeBolt, D. Ferguson, G. Seibel, and P. Kollman. Amber, a computer

program for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to elucidate the structures and energies of molecules. *Comp. Phys. Commun.*, 91:1–41, 1995.

[10] W. F. van Gunsteren and H. J. C. Berendsen. Computer simulation of molecular dynamics: Methodology, applications and perspectives in chemistry. *Angew. Chem. Int. Ed. Engl.*, 29:992–1023, 1990.

[11] W. B. Streett, D. J. Tildesley, and G. Saville. Multiple time step methods in molecular dynamics. *Mol. Phys.*, 35:639–648, 1978.

[12] C. S. Peskin and T. Schlick. Molecular dynamics by the backward: Euler's method. *Commun. Pure Appl. Math.*, 42:1001–1031, 1989.

[13] T. Schlick, E. Barth, and M. Mandziuk. Biomolecular dynamics at long timesteps: Bridging the timescale gap between simulation and experimentation. *Annu. Rev. Biophys. Biomol. Struct.*, 26:181–222, 1997.

[14] C. F. Sanz-Navarro and R. Smith. Numerical calculations using the hypermolecular dynamics method. *Comp. Phys. Comm.*, 137:206, 2001.

[15] A. Amadei, A. B. Linssen, and H. J. C. Berendsen. Essential dynamics of proteins. *Proteins*, 17:412–225, 1993.

[16] A. Krammer, H. Lu, B. Isralewitz, K. Schulten, and V. Vogel. Forced unfolding of the fibronectin type III module reveals a tensile molecular recognition switch. *Proc. Natl. Acad. Sci. USA*, 96:1351–1356, 1999.

[17] J. Schlitter, M. Engels, and P. Krüger. Targeted molecular dynamics: A new approach for searching pathways of conformational transitions. *J. Mol. Graphics*, 12:84–89, 1994.

[18] R. Böckmann and H. Grubmüller. Nanoseconds molecular dynamics simulations of primary mechanical energy transfer steps in F_1-ATP synthase. *Nat. Struc. Biol.*, 9:196–202, 2002.

[19] F. Noé, F. Ille, J. C. Smith, and S. Fischer. Automated computation of low-energy pathways for complex rearrangements in proteins: Application to the conformational switch of ras p21. *Proteins*, 59:534–544, 2005.

[20] H. Grubmüller. Predicting slow structural transitions in macromolecular systems: conformational flooding. *Phys. Rev. E*, 52:2893, 1995.

[21] S. Huo and J. E. Straub. Direct computation of long time processes in peptides and proteins: Reaction path study of the coil-to-helix transition in polyalanine. *Proteins*, 36:249–261, 1999.

[22] R. Czerminski and R. Elber. Self avoiding walk between two fixed points as a tool to calculate reaction paths in large molecular systems. *Int. J. Quant. Chem.*, 24:167, 1990.

[23] H. Jónsson, G. Mills, and K. W. Jacobsen. *Classical and Quantum Dynamics in Condensed Phase Simulations*, chapter Nudged Elastic Band Method for Finding Minimum Energy Paths of Transitions, pages 385–404. World Scientific, 1998.

[24] S. Fischer and M. Karplus. Conjugate peak refinement: an algorithm for finding reaction paths and accurate transition states in systems with many degrees of freedom. *Chem. Phys. Lett.*, 194:252–261, 1992.

[25] A. N. Bondar, M. Elstner, S. Suhai, J. C. Smith, and S. Fischer. Mechanism of primary proton transfer in bacteriorhodopsin. *Structure*, 12:1281–1288, 2004.

[26] A. D. Gruuia, A. N. Bondar, J. C. Smith, and S. Fischer. Mechanism of a molecular valve in the halorhodopsin chloride pump. *Structure*, 13:617–627, 2005.

[27] S. Fischer, S. Michnick, and M. Karplus. A mechanism for rotamase catalysis by the fk506 binding protein (fkbp). *Biochemistry*, 32:13830–13837, 1993.

[28] F. H. Stillinger and T. A. Weber. Hidden structure in liquids. *Physical Reviews A*, 25:978–989, 1982.

[29] F. H. Stillinger. A topographic view of supercooled liquids and glass formation. *Science*, 267:1935–1939, 1995.

[30] R. Czerminski and R. Elber. Reaction path study of conformational transitions and helix formation in a tetrapeptide. *Proc. Nat. Acad. Sci. USA*, 86:6963–6967, 1989.

[31] R. Czerminski and R. Elber. Reaction path study of conformational transitions in flexible systems: Application to peptides. *J. Chem. Phys.*, 92:5580–5601, 1990.

[32] R. S. Berry and R. Breitengraser-Kunz. Topography and dynamics of multidimensional interatomic potential surfaces. *Phys. Rev. Lett.*, 74:3951–3954, 1995.

[33] D. J. Wales. Structure, Dynamics, and Thermodynamics of Clusters: Tales from Topographic Potential Surfaces. *Science*, 271:925–933, 1996.

[34] K. D. Ball, R. S. Berry, R. E. Kunz, F.-Y. Li, A. Proykova, and D. J. Wales. From topographies to dynamics on multidimensional potential energy surfaces of atomic clusters. *Science*, 271:963–967, 1996.

[35] O. M. Becker and M. Karplus. The topology of multidimensional potential energy surfaces: Theory and application to peptide structure and kinetics. *Journal of Chemical Physics*, 106:1495–1517, 1996.

[36] Y. Levy and O. M. Becker. Effect of conformational constraints on the topography of complex potential energy surfaces. *Phys. Rev. Lett.*, 81:1126–1132, 1998.

[37] M. A. Miller and D. J. Wales. Energy landscape of a model protein. *J. Chem. Phys.*, 111(14):6610–6616, 1999.

[38] P. N. Mortenson and D. J. Wales. Energy landscapes, global optimization and dynamics of the polyalanine $Ac(ala)_8NHMe$. *Journal of Chemical Physics*, 114:6443–6453, 2001.

[39] Y. Levy and O. M. Becker. Energy landscapes of conformationally constrained peptides. *Journal of Chemical Physics*, 114:993–1009, 2001.

[40] C. L. Brooks III, J. N. Onuchic, and D. J. Wales. Taking a walk on a landscape. *Science*, 293:612–613, 2001.

[41] Y. Levy, J. Jortner, and O. M. Becker. Dynamics of hierarchical folding on energy landscapes of hexapeptides. *J. Chem. Phys.*, 115:10533–10547, 2001.

[42] Y. Levy, J. Jortner, and O. M. Becker. Solvent effects on the energy landscapes and folding kinetics of polyalanines. *Proc. Nat. Acad. Sci. USA*, 98:2188–2193, 2001.

[43] P. N. Mortenson, D. A. Evans, and D. J. Wales. Energy landscapes of model polyalanines. *J. Chem. Phys.*, 117:1363–1376, 2002.

[44] Y. Levy and O. M. Becker. Conformational polymorphism of wild-type and mutant prion proteins: energy landscape analysis. *Proteins*, 47:458–468, 2002.

[45] D. A. Evans and D. J. Wales. Free energy landscapes of model peptides and proteins. *J. Chem. Phys.*, 118:3891–3897, 2003.

[46] D. A. Evans and D. J. Wales. The free energy landscape and dynamics of met-enkephalin. *J. Chem. Phys.*, 119:9947–9955, 2003.

[47] D. J. Wales and J. P. K. Doye. Stationary points and dynamics in high-dimensional systems. *J. Chem. Phys.*, 119:12409–12416, 2003.

[48] D. A. Evans and D. J. Wales. Folding of the gb1 hairpin peptide from discrete path sampling. *J. Chem. Phys.*, 121:1080–1090, 2004.

[49] F. Despa, D. J. Wales, and R. S. Berry. Archetypal energy landscapes: Dynamical diagnosis. *J. Chem. Phys.*, 122:024103, 2005.

[50] Y. Levy, J. Jortner, and O. M. Becker. Dynamics of hierarchical folding on energy landscapes of hexapeptides. *J. Chem. Phys.*, 115:10533–10547, 2001.

[51] M. A. Miller, J. P. K. Doye, and D. J. Wales. Energy landscapes of model polyalanines. *J. Chem. Phys.*, 117:1363–1376, 2002.

[52] D. J. Wales. Discrete path sampling. *Mol. Phys.*, 100:3285–3305, 2002.

[53] M. S. Apaydin, D. L. Brutlag, C. Guestrin, D. Hsu, J.-C. Latombe, and C. Varma. Stochastic Roadmap Simulation: An Efficient Representation and Algorithm for Analyzing Molecular Motion. *J. Comp. Bio.*, 10:257–281, 2003.

[54] E. Dijkstra. A note on two problems in connexion with graphs. *Num. Math.*, 1:269–271, 1959.

[55] P. Kollmann. Free energy calculations: Applications to chemical and biochemical phenomena. *Chem. Rev.*, 93:2395–2417, 1993.

[56] I. Horenko, E. Dittmer, A. Fischer, and Ch. Schütte. Automated model reduction for complex systems exhibiting metastability. *SIAM Mult. Mod. Sim.*, 2005, submitted.

[57] W. J. Moore and D. O. Hummel. *Physikalische Chemie*. Walter de Gruyter, Berlin, 3rd edition, 1983.

[58] G. Henkelman, G. Jóhannesson, and H. Jónsson. *Progress on Theoretical Chemistry and Physics*, chapter Methods for Finding Saddle Points and Minimum Energy Paths, pages 269–300. Kluwer Academic Publishers, 2000.

[59] W. E, W. Ren, and E. Vanden-Eijnden. String method for the study of rare events. *Phys. Rev. B*, 66:052301, 2002.

[60] A. Tournier and J.C. Smith. Principal components of the protein dynamical transition. *Physical Review Letters*, 91:208106, 2003.

[61] R. Elber. *Recent Developments in Theoretical Studies of Proteins*, chapter Reaction Path Studies of Biological Molecules. World Scientific, Singapore, 1996.

[62] E.B. Wilson, J.C. Decius, and P.C. Cross. *Molecular Vibrations*. McGraw-Hill, New York, 1955.

[63] D. Perahia and L. Mouawad. Computation of low-frequency normal modes in macromolecules: improvements to the method of diagonalization in a mixed basis and application to hemoglobin. *Comput. Chem.*, 19:241–246, 1995.

[64] I. N. Bronstein, K. A. Semendjajew, G. Musiol, and H. Mühlig. *Taschenbuch der Mathematik*. Harri Deutsch, Frankfurt, 5th edition, 2001.

[65] G. M. Torrie and J. P. Valleau. Nonphysical Sampling Distributions in Monte Carlo Free-Energy Estimation: Umbrella Sampling. *J. Comp. Phys.*, 23:187–199, 1977.

[66] R. Rajamani, K. J. Naidoo, and J. Gao. Implementation of an adaptive umbrella sampling method for the calculation of multidimensional potential of mean force of chemical reactions in solution. *Proteins*, 24:1775–1781, 2003.

[67] M. Berkowitz, J. D. Morgan, J. A. McCammon, and S. H. Northrup. Diffusion-controlled reactions: A variational formula for the optimum reaction coordinate. *J. Chem. Phys*, 79:5563–5565, 1983.

[68] S. Huo and J. E. Straub. The maxflux algorithm for calculating variationally optimized reaction paths for conformational transitions in many body systems at finite temperature. *J. Chem. Phys.*, 107:5000–5006, 1997.

[69] R. E. Bellmann. On a routing problem. *Appl. Math.*, 16:87–90, 1958.

[70] L.R. Ford Jr. Network flow theory. Paper P-923, The RAND Corperation, Santa Moncia, California, August 1956.

[71] D. Frigioni and A. Marchetti-Spaccamela. Fully dynamic algorithms for maintaining shortest paths trees. *Journal of Algorithms*, pages 251–281, 2000.

[72] D. Eppstein. Finding the k shortest paths. *Proc. 35th Symp. Found. Comp. Sci.*, pages 154–165, 1994.

[73] H. Nagamochi and T. Ibaraki. Computing edge connectivity in multigraphs and capacitated graphs. *SIAM Journal on Discrete Mathematics*, 5:54–66, 1992.

[74] J. Bang-Jensen and G. Gutin. *Digraphs.* Springer-Verlag UK, 2002.

[75] D. Hoffmann and E.-W. Knapp. Polypeptide folding with off-lattice Monte-Carlo dynamics: the method. *Eur. Biophysics J.*, 24:387–404, 1996.

[76] M. Mezei. Efficient Monte Carlo sampling for long molecular chains using local moves, tested on a solvated lipid bilayer. *J. Chem. Phys*, 118:3874–3879, 2003.

[77] D. R. Lowy and B. M. Willumsen. Function and regulation of ras. *Ann. Rev. Biochem.*, 62:851–891, 1993.

[78] E. F. Pai, U. Krengel, G. A. Petsko, R. S. Goody, W. Kabsch, and A. Wittinghofer. Refined crystal structure of the triphosphate conformation of H-ras p21 at 1.35 Å resolution: implications for the mechanism of GTP hydrolysis. *EMBO J.*, 9:2351, 1990.

[79] L. A. Tong, A. de Vos, M. V. Milburn, and S. H. Kim. Crystal structures at 2.2 Å resolution of the catalytic domains of normal ras protein and an oncogenic mutant complexed with GDP. *J. Mol. Biol.*, 217:503, 1991.

[80] S. E. Neal, J. F. Eccleston, and M. R. Webb. Hydrolysis of GTP by $p21^{NRAS}$, the *NRAS* protooncoge ne product, is accompanied by a conformational change in the wild-type protein. Use of a single fluorescent probe at the catalytic site. *Proc. Natl. Acad. Sci. USA*, 87:3562–3565, 1990.

[81] J. Ma and M. Karplus. Molecular switch in signal transduction: Reaction paths of the conformational changes in ras p21. *Proc. Natl. Acad. Sci. USA*, 94:11905–11910, 1997.

[82] J. F. Díaz, B. Wroblowski, J. Schlitter, and Y. Engelborghs. Calculation of Pathways for the Conformational Transition Between the GTP- and GDP-Bound States of the Ha-ras-p21 Protein: Calculations With Explicit Solvent Simulations and Comparison With Calculations in Vacuum. *Proteins*, 28:434–451, 1997.

[83] M. Schaefer and M. Karplus. A Comprehensive Analytical Treatment of Continuum Electrostatics. *J. Chem. Phys.*, 100:1578–1599, 1996.

[84] H. Hotelling. Analysis of a complex of statistical variables into principal components. *J. Edu. Psych.*, 24:417–441, 1933.

[85] J. W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Trans. Comp.*, C-18(5):401–409, 1969.

[86] R. Preis, M. Dellnitz, M. Hessel, Ch. Schütte, and E. Meerbach. Dominant paths between almost invariant sets of dynamical systems. *Preprint 154*, 2004.

[87] J. M. Carr, S. A. Trygubenko, and D. J. Wales. Finding pathways between distant local minima. *J. Chem. Phys.*, 122:234903, 2005.

[88] I. R. Vetter, A. Arndt, U. Kutay, D. Goerlich, and A. Wittinghofer. Structural view of the Ran-Importin beta interaction at 2.3 Å resolution. *Cell*, 97:635–646, 1999.

[89] M. Kjeldgaard, J. Nyborg, and B. F. C. Clark. The gtp binding motif: variations on a theme. *FASEB J.*, 10:1347–1368, 1996.

[90] H. Frauenfelder, S. G. Sligar, and P. G. Wolynes. The energy landscapes and motions of proteins. *Science*, 254:1598–1603, 1991.

[91] D. C. Sullivan and I. D. Kuntz. Conformation Spaces of Proteins. *Proteins*, 42:495–511, 2001.

[92] C. A. R. Hoare. Quicksort. *Comp. J.*, 5:10–15, April 1962.

[93] M. Karoński and A. Ruciński. The origins of the theory of random graphs. In R. L. Graham and J. Nešetřil, editors, *The mathematics of Paul Erdös*, volume 13 of *Algorithms and Combinatorics*, pages 311–336. Springer, Berlin, 1997.

[94] E. Neria, S. Fischer, and M. Karplus. Simulation of activation free energies in molecular systems. *J. Chem. Physics*, 105:1902–1921, 1996.